nexMart
That's E-Business.

User Manual

MARTSERVICES

Specifications and technical
implementation in IT-Systems

Please note that there is no guarantee for technical details or the applicability of software referring the product description and the application documentation.

The documentation referring the product of the contract is also stored on the computer centre from nexMart and can be requested and printed at any time.

Copyright

The information given in the document is under copyright. It is forbidden to copy, modify, translate, microfilming, save on electronic medias as well as recreate the document or parts of the document electronically – this right is reserved just for nexMart GmbH & Co. KG. Software referring the documentation is property of the Copyright-owner.

Copyright © 2013 by nexMart GmbH & Co. KG, Stuttgart. All rights reserved.

Author: nexMart GmbH & Co. KG · Gropiusplatz 10 · 70563 Stuttgart

Date	Version	Author	Change
01.01.2011	1.4	Tony Bußler nexMart GmbH	Translation on the bases of the german original version 1.4

Table 1 *Proof of change*

Index

Index 3

1. What are the MartServices? 6
   1.1. What do MartServices? 6
   1.2. Target Group 6
   1.3. Added Value 6

2. Technical Devices 7
   2.1. Construction of the individual MartServices 7
   2.2. Definition MartLink 7
   2.3. Definition WebService 7
   2.4. System and Operating Conditions 7

3. Base of all MartLinks 8
   3.1. URL 9
   3.2. Passing of Request Parameter 9
   3.3. Passing of Response Parameter 9
   3.4. Transfer Method 10
   3.5. Error Reports 10
   3.6. International Use 10
   3.7. Security 11
   3.8. Version Management 11
   3.9. Traceability of the Response 11
   3.10. Necessary Parameter for all MartLinks 11
      3.10.1. Request 12
      3.10.2. Response 13

4. Product Details 14
   4.1. Function 14
   4.2. Key Features 14
   4.3. Customer Benefits 14
   4.4. What details are displayed? 14
   4.5. Integration into the IT 16
   4.6. Technical Implementation 16
      4.6.1. URL 16
      4.6.2. Request 17
      4.6.3. Response 17
      4.6.4. Multiple Hits in Search 17
      4.6.5. No Hit in Search 18

5. Takeover of Shopping Baskets 19
5.1. Function 19
5.2. Entry Point into the Portal 19
5.3. Fill the Shopping Basket 21
5.4. Takeover of Shopping Baskets 22
5.5. Already created Shopping Baskets 24
5.6. Technical Implementation 24
5.7. MartLink PickBasket 25
5.7.1. URL 25
5.7.2. Request 25
5.7.3. Response 25
5.8. WebService GetBasket 27
5.8.1. URL 27
5.8.2. Format of the Shopping Basket 27
5.8.3. Exceptional Act 27
5.8.4. Request Parameter 28
5.8.5. Response Parameter 29
5.8.6. Results 32

6. Order Centre 33
6.1. Function 33
6.2. Entry Point 33
6.3. Search and Filtern 34
6.4. Differences to the Online Features 34
6.5. Technische Implementierung 35
6.5.1. URL 35
6.5.2. Request 35
6.5.3. Response 36

7. Order Tracking 37
7.1. Function 37
7.2. Search and Filter 38
7.3. Technical Implementatian 38
7.3.1. URL 38
7.3.2. Request 39
7.3.3. Response 39

8. Product Configuration 40
8.1. Function 40
8.2. Configuration of Products 40
8.3. Product Colors 40
8.4. Technical Implemtation 43
8.4.1. URL 43
8.4.2. Request 44
8.4.3. Response 45
8.5. Advanced Settings 46

9. MartLink AutoLogin 47

10. Prices and Availability 49
10.1. Function 49
10.2. Prices und Availabilities in the own Shopping Basket 49
10.2.1. Paractical Example with Srceenshots 49
10.2.2. Attributes/Process 50
10.2.3. Technical Implementation 51
10.3. View Prices and Availability 54
10.3.1. Practical Examples with Screenshots 54
10.3.2. Attributes/Process 55
10.3.3. Technical Implementation 58
10.4. Online Request of Price and Availability 60

11. Base of all WebServices 61
11.1.1. Request (for new structure) 61
11.1.2. Response (for new structur) 61

12. WebService ProcessOrder 62
12.1. Service-Routing 63
12.2. URL 63
12.3. International Use 63
12.4. Security 63
12.5. Format der Bestellung 64
12.6. Client Information 64
12.7. Version Management 64
12.8. Traceability of the Response 64
12.9. Exceptions and Error Handling 65
12.10. Technische Implementierung 66
12.10.1. Request Parameter 66
12.10.2. Response 67
12.10.3. Results 69

13. FAQ 70

14. Appendix 71
14.1. Error Codes 71
14.2. VBA Example Shopping Basket Takeover 79
14.3. VBA Example ProcessOrder 84

1. What are the MartServices?

1.1. What do MartServices?

Enterprise resource planning systems can be enriched with portal functions by MartServices. The functions of the portal are implemented directly into the computer of the customer. The use of the portal functionality expands each ERP-System to better product and shipping information and more efficient purchasing processes. Users of the MartServices don't need to change their applications and they have not to log in separately to the portal.

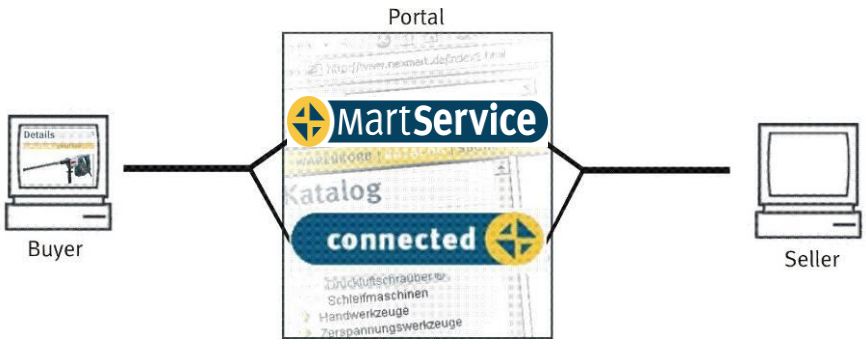

Figure 1 *Schematic representation of MartServices*

1.2. Target Group

The target groups are providers of application software and system vendors, as well as purchasers and contract agents who want to shop online via nexMart and use the portal functionality in familiar internal software.

1.3. Added Value

- Integration of portal functionality in application software
- Enrichment of the user software to better information and more efficient purchasing processes
- Using the features without portal Login or the change of applications

2. Technical Devices

2.1. Construction of the individual MartServices

MartServices are composed of individual technology components. These devices are different WebServices or MartLinks, sometimes a combination of both.

2.2. Definition MartLink

In a mart link pages of the portal are accessed and appear in the external application programs. Information of the portal and portal features can be used by MartLinks. When dealing with features no data in the application program are stored, but everything remains on the portal.

2.3. Definition WebService

Data can be loaded into the application programs by a WebService. Because of the WebService an application and the portal are able to communicate with each other. That means besides of using the portal features, the user may also save his work in its own application.

2.4. System and Operating Conditions

Users and their respective suppliers have to be enabled for using the services.

The user program has to be compatible with a browser to view portal features.

3. Base of all MartLinks

A MartLink provides a direct entry into the portal or commands a service of the portal. This is accessed via an http(s) Post-Request that is answered by an html page.

URL Structure

The MartLink is established according to the following pattern:

http://‹PORTAL›/martlink/‹SERVICE›,

‹PORTAL› is the URL of the portal und ‹SERVICE› the name of the MartLinks or WebServices.

Illustration

http://www.portal.de/martlink/productdetail

Parameter

There are parameters which apply to all MartLinks and have to be implemented, as well as specific parameters which are needed only for the particular MartLinks.

In Chapter 3 are all request and response parameters listed which apply to all MartLinks alike. In addition, the individual parameters have to be considered. These are listed in the individual chapters.

Illustration

An authentication and authorization of the user and the order number or GTIN (EAN) code of the device is required in the case of the MartLink ProductDetail. For example, the parameter “ProductID” and a String as the value is sent in the http Post-Request.

3.1. URL

The URL for the http(s) Protocol of a MartLink is established according to the following pattern:

http://‹Portal-URL›/martlink/‹MartLink-Id›

or

https://‹Portal-URL›/martlink/‹MartLink-Id›

‹Portal-URL› is the base URL of the portal and ‹MartLink-Id› is the identifier of the called MartLink.

e.g.:

a Get call with a HTML Link

‹a href ="http://demomart.nexmart.de/martlink/viewbasket?

BuyerOrganization=BDE……&

BuyerUserName=name.surname&

…

BasketID=7uoKEBhjJ10AAAENzQS.KMUk"›

ViewBasket

‹/a›

3.2. Passing of Request Parameter

The parameters for calling a MartLink are passed in http(s)-Request, using POST. Please keep attention to capitalization of the parameter. An error-page will occur, if invalid parameters prevent the call of the first page of the MartLink.

3.3. Passing of Response Parameter

The result of a MartLink is initiated by the last user action and transmits the corresponding response parameters to the client from the **ClientTargetURL**.

The response can be transmitted by four different methods to the client. These are specified in the ClientRequestMethode.

3.4. Transfer Method

localget:

A response page is transmitted to the client by means of a redirect. The specified URL of the ClientRequestURL is transmitted with the parameter GET during the loading process. The general request parameters are not transmitted to the client.

Localpost:

The localpost method works like the localget method, except that the data are transmitted via POST to the client and the general request parameters are included.

remoteget:

The remoteget method transmits the parameter directly to the client via GET. That is only possible if the client is directly accessible from outside. The general parameters are not transmitted.

remotepost:

The remotepost method transmits the parameters directly to the client, just like the remoteget method. However, the general parameters will be transmitted in contrast to the remoteget method.

The GET method doesn't transfer the general MartLink Response parameter because of the limited number of possible characters (e.g. 2048 for Microsoft Internet Explorer).

3.5. Error Reports

An error may occur because of an invalid authentication, authorization or invalid parameters in the Request. If an error occurs there will be an error page with a list of missing parameters.

3.6. International Use

Following parameters have to be submitted for the internationalization of the MartLinks:

» **MartLinkLocale:**
MartLinkLocal is a valid local ID of 4 letters. The first 2 letters define the language according to ISO 639 and the last 2 letters define the country according to ISO 3166.

» **MartLinkCurrency:**
MartLinkCurrency is a valid currency ID of 3 letters according to ISO 4217.

3.7. Security

The transmission path for the request needs to be secured. Also the authentication and authorization has to be ensured for the caller to guarantee the security of the MartLink.

The security of the transmission path is given by the use of the **http(s)-protocol**. The parameters of the transmission will be left unprotected by the use of http-protocol. Therefore, we recommend the use of the http(s)-protocol.

The user/password combination is transmitted for authentication. The user is identified by **BuyerOrganization** and **BuyerUserName** and authorized by **BuyerUserPassword**. Both have to match an activated and valid ID on the portal.

3.8. Version Management

MartLinks will be further developed. The version of the MartLink (MartLinkVersion) is indicated in the request to run and route one and the same MartLink parallel.

3.9. Traceability of the Response

The information of a MartLink for tracing is provided in the response. This information could be used for debugging.

The following information is provided:

- **PortalID:**
  The portal identification which is generated by the Response.
- **HostName:**
  The name of the host which is generated by the response. This information is used only for internal use and tracking errors.
- **ResponseTimeStamp:**
  A time stamp that holds the date and time of the created Response.

3.10. Necessary Parameter for all MartLinks

! **When constructing MartLinks keep attention to capitalization.**

3.10.1. Request

Name	Type	Required/ Optional	Description
ClientName	String [1..128] case sensitive	Required	The name of the calling client.
ClientTargetURL	String [1..512] case insensitive	Optional	URL of the client to whom the Request is sent.
ClientRequestMethod	String [1..32] [“localget”, ”localpost”]	Required	Method by which the Response to the calling Client is passed.
MartLinkId	String [1..128] case sensitive	Required	Identifier of the calling MartLinks. e.g. ViewBasket
MartLinkVersion	String [1..32] case insensitive	Required	Version of the MartLink, named in MartLinkId; currently 1.0 (as of 21.12.2009)
MartLinkPortalID	String [1...128] case insensitive	Required	Name of the portal.
MartLinkSupplierId	String [0..128] case insensitive	Optional	A valid identifier of a supplier.
MartLinkLocale	String [5] case sensitive	Optional	A valid locale-ID consists three parts: - ISO 639 Language Code 2 letter lower case - separator „_“ - ISO 3166 Country Code 2 letter upper case e.g. de_DE, de_AT, en_GB, ...
MartLinkCurrency	String [3] case insensitive	Optional	A valid currency-ID according to ISO 4217. e.g. EUR, CHF, USD, etc.
BuyerOrganization	String [9]	Required	ID of the customers company consist 3 letters and 6 digits:

Name	Type	Required/ Optional	Description
	case sensitive		e.g. BDE999999.
BuyerUserName	String [1..255] case sensitive	Required	The username of the customer consists of ‹name›.‹surname›. Special characters are not allowed.
BuyerUserPassword	String [8..128] case sensitive	Required	Password of the user.

Table 1 *General Request parameter of a MartLink*

3.10.2. Response

Name	Type	Guaranteed?	Description
PortalID	String [1..128] case insensitive	Yes	Identifier of the portal which generated the Response.
HostName	String [1..256] case sensitive	No	Name of the System which generated the Response. This information is used only for internal use and tracking errors.
ResponseTimeStamp	DateTime	No	A time stamp that holds the date and time of the created Response.

Table 2 *General Response parameter of a MartLink*

4. Product Details

4.1. Function

Product and delivery information can be accessed by your own ERP. There is no need of a portal login. In this way your usual computer system can be enriched by numerous information of products from all suppliers of the portal.

4.2. Key Features

- Link to product details from an external system, e.g. ERP-System
- Call of product details of a supplier, e.g. spare parts, accessories, user manuals, etc.
- Show the product details in a browser window via URL or integrated in the computer interface.
- This feature can be activated for every supplier und customer.

4.3. Customer Benefits

- The user obtains detailed information on products without a portal login or leaving his usual program.
- Efficient call of product details without time-consuming search.
- Extensive product details for users who do not order via internet browser.
- The user receives all the information of a product quick and easy, except it is not offered by the seller on the portal.

4.4. What details are displayed?

Depending on the database, all product information is available, except it is not offered by the seller.

It could be:

- Details to the article
  (Product description, commercial and technical data)
- Photos of products and usage
- Spare parts
- Accessories
- Variations
- User manuals

» Dimensioned and exploded drawings
» Other datasheets

Prices will not be displayed!

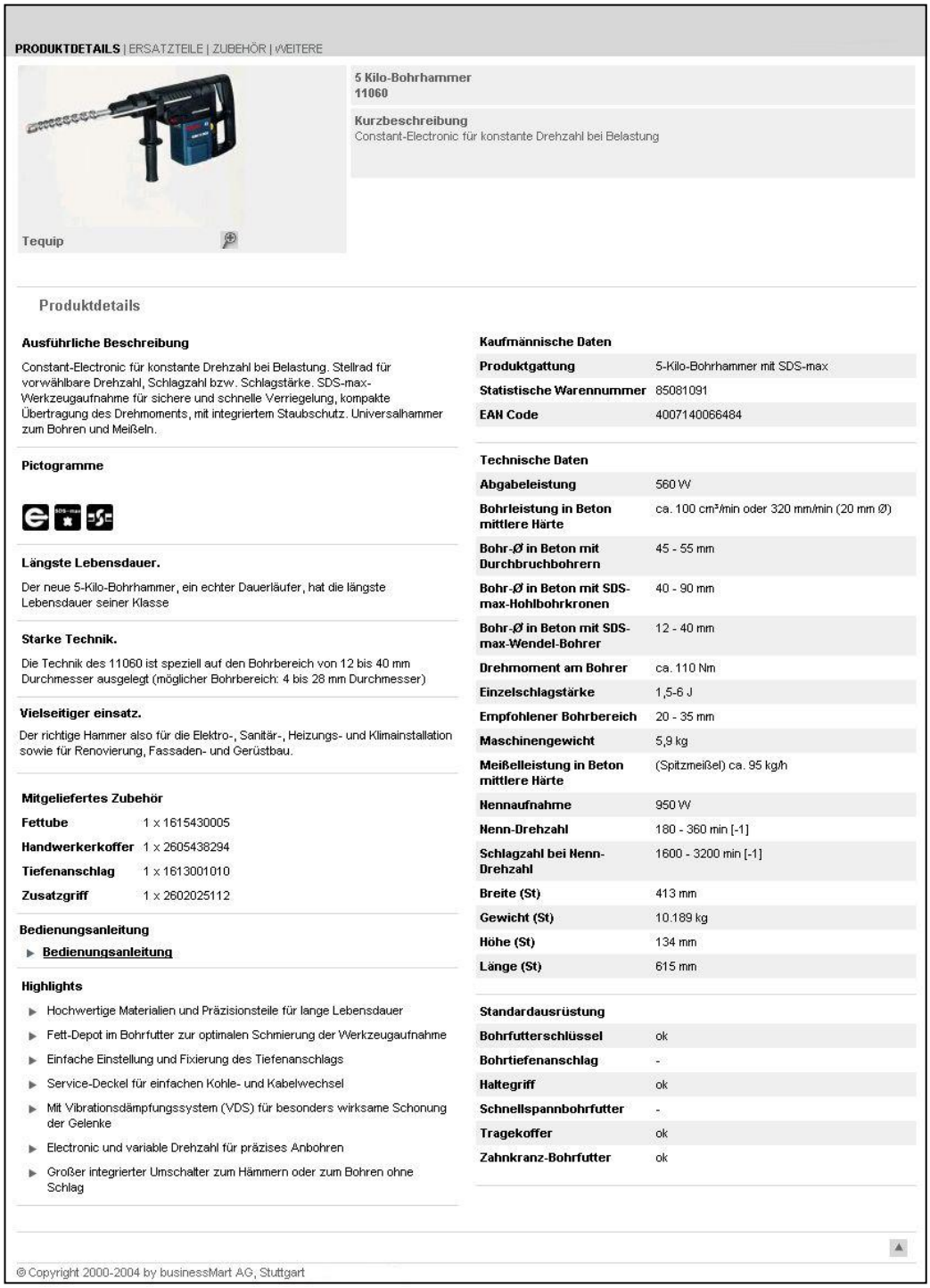
PRODUKTDETAILS | ERSATZTEILE | ZUBEHÖR | WEITERE
5 Kilo-Bohrhammer
11060
Kurzbeschreibung
Constant-Electronic für konstante Drehzahl bei Belastung
Tequip
Produktdetails
Ausführliche Beschreibung
Constant-Electronic für konstante Drehzahl bei Belastung. Stellrad für vorwählbare Drehzahl, Schlagzahl bzw. Schlagstärke. SDS-max-Werkzeugaufnahme für sichere und schnelle Verriegelung, kompakte Übertragung des Drehmoments, mit integriertem Staubschutz. Universalhammer zum Bohren und Meißeln.
Pictogramme
Längste Lebensdauer.
Der neue 5-Kilo-Bohrhammer, ein echter Dauerläufer, hat die längste Lebensdauer seiner Klasse
Starke Technik.
Die Technik des 11060 ist speziell auf den Bohrbereich von 12 bis 40 mm Durchmesser ausgelegt (möglicher Bohrbereich: 4 bis 28 mm Durchmesser)
Vielseitiger einsatz.
Der richtige Hammer also für die Elektro-, Sanitär-, Heizungs- und Klimainstallation sowie für Renovierung, Fassaden- und Gerüstbau.
Mitgeliefertes Zubehör
Fettube 1 x 1615430005
Handwerkerkoffer 1 x 2605438294
Tiefenanschlag 1 x 1613001010
Zusatzgriff 1 x 2602025112
Bedienungsanleitung
Bedienungsanleitung
Highlights
Hochwertige Materialien und Präzisionsteile für lange Lebensdauer
Fett-Depot im Bohrfutter zur optimalen Schmierung der Werkzeugaufnahme
Einfache Einstellung und Fixierung des Tiefenanschlags
Service-Deckel für einfachen Kohle- und Kabelwechsel
Mit Vibrationsdämpfungssystem (VDS) für besonders wirksame Schonung der Gelenke
Electronic und variable Drehzahl für präzises Anbohren
Großer integrierter Umschalter zum Hämmern oder zum Bohren ohne Schlag
Kaufmännische Daten
Produktgattung 5-Kilo-Bohrhammer mit SDS-max
Statistische Warennummer 85081091
EAN Code 4007140066484
Technische Daten
Abgabeleistung 560 W
Bohrleistung in Beton mittlere Härte ca. 100 cm³/min oder 320 mm/min (20 mm Ø)
Bohr-Ø in Beton mit Durchbruchbohrern 45 - 55 mm
Bohr-Ø in Beton mit SDS-max-Hohlbohrkronen 40 - 90 mm
Bohr-Ø in Beton mit SDS-max-Wendel-Bohrer 12 - 40 mm
Drehmoment am Bohrer ca. 110 Nm
Einzelschlagstärke 1,5-6 J
Empfohlener Bohrbereich 20 - 35 mm
Maschinengewicht 5,9 kg
Meißelleistung in Beton mittlere Härte (Spitzmeißel) ca. 95 kg/h
Nennaufnahme 950 W
Nenn-Drehzahl 180 - 360 min [-1]
Schlagzahl bei Nenn-Drehzahl 1600 - 3200 min [-1]
Breite (St) 413 mm
Gewicht (St) 10.189 kg
Höhe (St) 134 mm
Länge (St) 615 mm
Standardausrüstung
Bohrfutterschlüssel ok
Bohrtiefenanschlag -
Haltegriff ok
Schnellspannbohrfutter -
Tragekoffer ok
Zahnkranz-Bohrfutter ok
© Copyright 2000-2004 by businessMart AG, Stuttgart

Figure 2 *Example for product details*

4.5. Integration into the IT

The portal feature “Product Details” can be placed anywhere in a computer system but that depends on the developers of the computer systems.

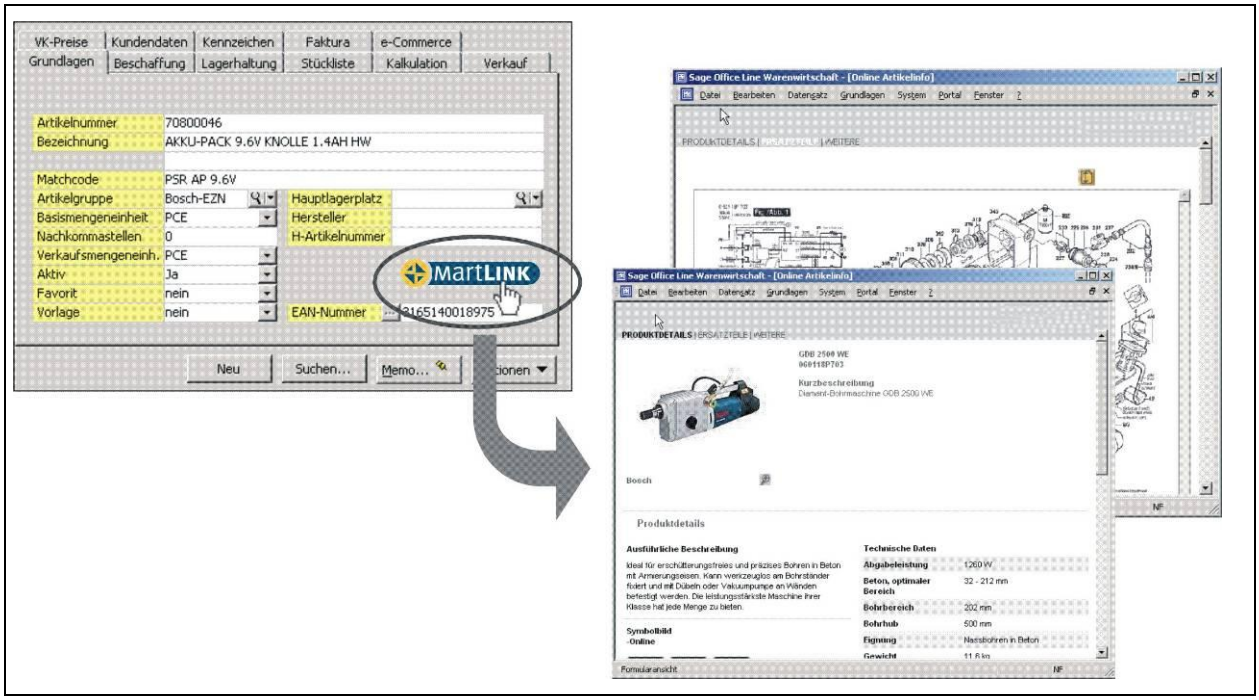

Figure 3 *Example for integrated product details displayed in a Popup window*

4.6. Technical Implementation

4.6.1. URL

The MartLink-ID for product details is “productdetail”.

https://‹Portal-URL›/martlink/ productdetail

In addition a target of the entry point can be provided. In that way it is possible to have a direct access to the spare part page, accessory page or other pages.

The MartLink ProductDetail does not return any parameters and therefore no Response.

4.6.2. Request

The following parameters in the table apply for the MartLink ProductDetail, as well as all general parameters from chapter 3.

Name	Type	Required/ Optional	Description
SupplierArticle Number	String [1..128] case sensitive	Required	Unique article number of the product given by the supplier. Mostly it is the order number or the GTIN (former EAN) Code.
TargetPage	String [“productdetail”, “spareparts”, “accessories”, “other”]	Optional	The appropriate tab is activated and the called page displayed if the requested information is available, otherwise the main page of the product (productdetail) is displayed.
MartLinkSupplierId	String [0..128] case insensitive	Required	A valid identifier of a supplier who is part of the portal.

Table 3 *Request parameter for the MartLink product details*

4.6.3. Response

The MartLink ProductDetail has no Response parameter. It is called only one side of the portal.

4.6.4. Multiple Hits in Search

A hit list will show up if there is an article number for two or more articles.

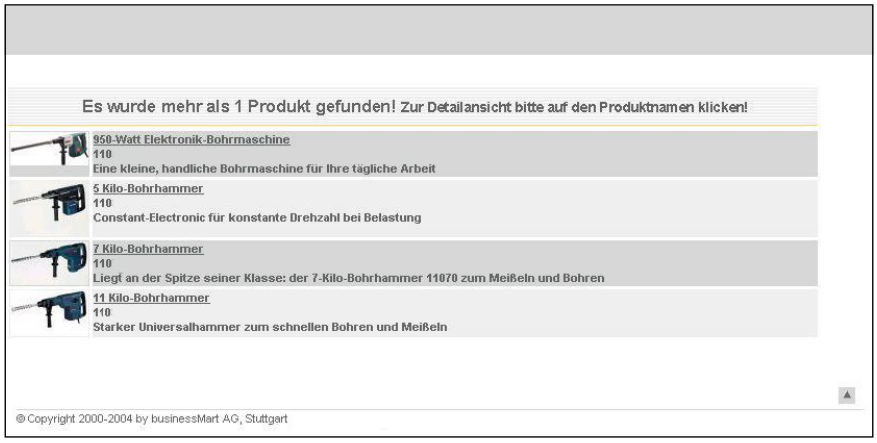
Es wurde mehr als 1 Produkt gefunden! Zur Detailansicht bitte auf den Produktnamen klicken!
950-Watt Elektronik-Bohrmaschine
110
Eine kleine, handliche Bohrmaschine für Ihre tägliche Arbeit
5 Kilo-Bohrhammer
110
Constant-Electronic für konstante Drehzahl bei Belastung
7 Kilo-Bohrhammer
110
Liegt an der Spitze seiner Klasse: der 7-Kilo-Bohrhammer 11070 zum Meißeln und Bohren
11 Kilo-Bohrhammer
110
Starker Universalhammer zum schnellen Bohren und Meißeln
© Copyright 2000-2004 by businessMart AG, Stuttgart

Figure 4 *Displayed list by multiple hits in search*

4.6.5. No Hit in Search

If the product is not available on the portal, the following message is displayed.

Figure 5 *No hit in search*

5. Takeover of Shopping Baskets

5.1. Function

The function “Takeover of Shopping Baskets” allows to fill a shopping basket on the portal interface and then to transfer to the disposition of the own IT-System.

All existing portal functions for filling a shopping basket are available for the customer, e.g. the product catalogues of the supplier and the product search.

After filling, the shopping basket is transferred directly into the IT-System or is saved as an open shopping basket. The open shopping basket can be picked up later.

The user books the shopping basket on its own program.

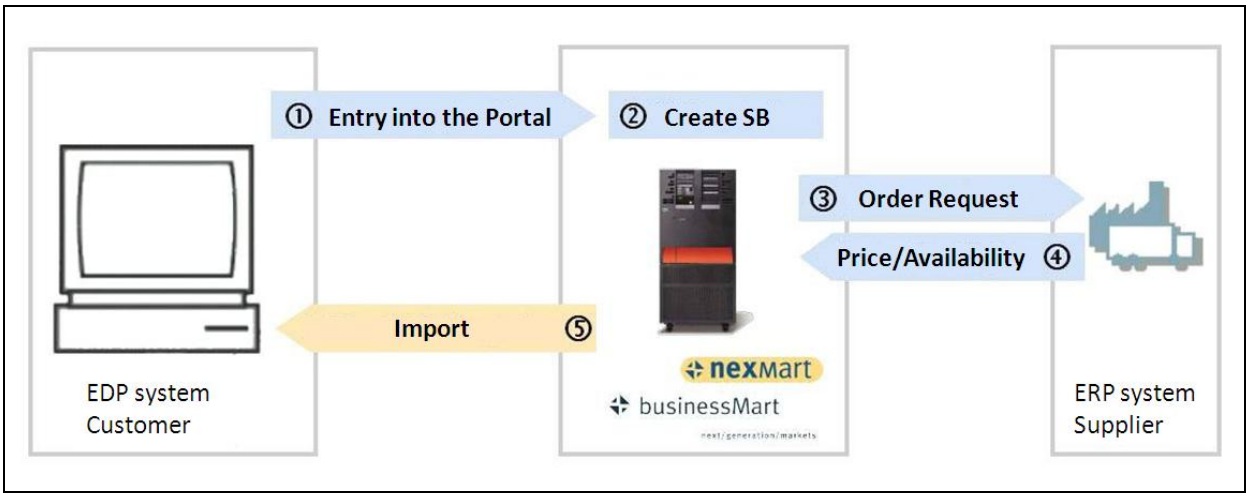

Figure 6 *Schematic representation of the process*

5.2. Entry Point into the Portal

The composition of the shopping baskets can be done from different entry points in the portal:

Entry Point into the Catalogue

The user can browse for products, access product details and place the product in the shopping basket by using the entry point.

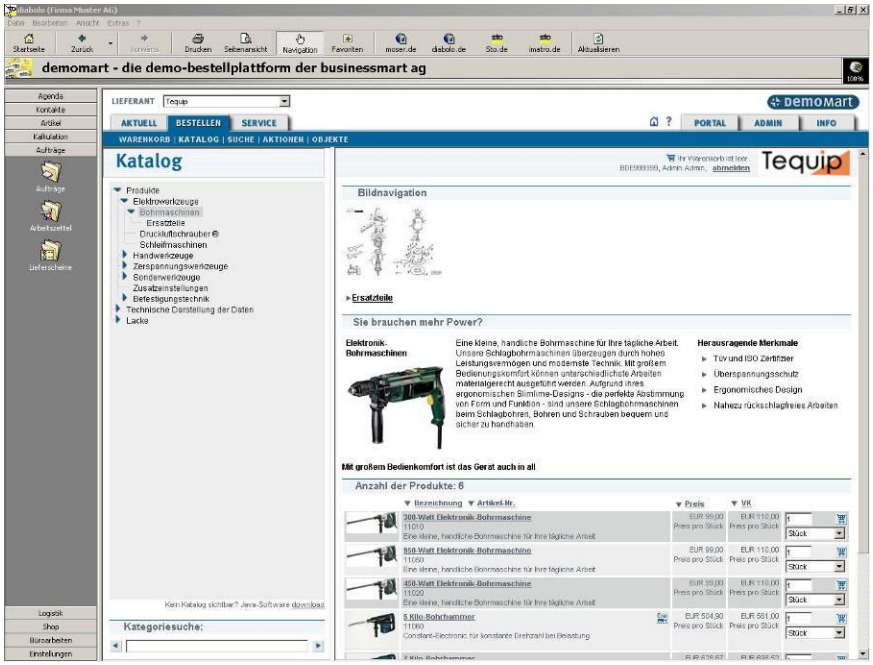

Figure 7 *Entry Point into the Catalogue*

Entry Point into the Product Search page

The product search is also available and the user can access it at any time.

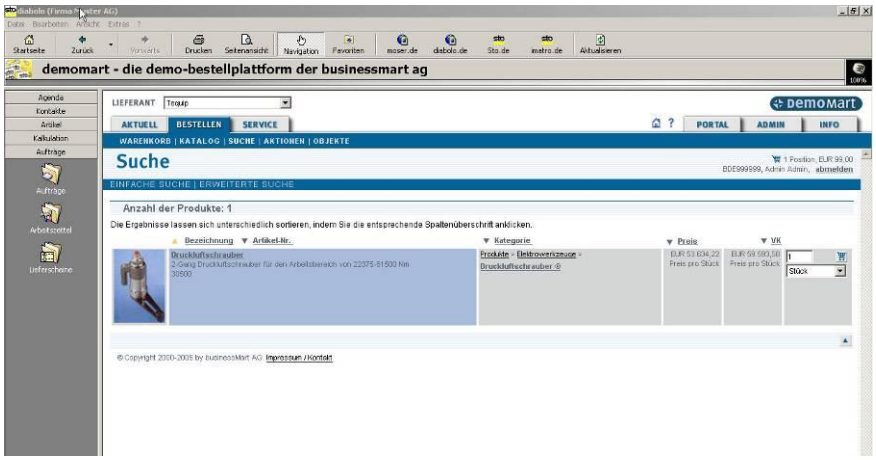

Figure 8 *Entry Point into the Product Search page*

Entry Point into the Shopping Basket

It is an additional opportunity to jump directly into a new shopping basket. At that point, the user can enter the article numbers in the fast entry lines.

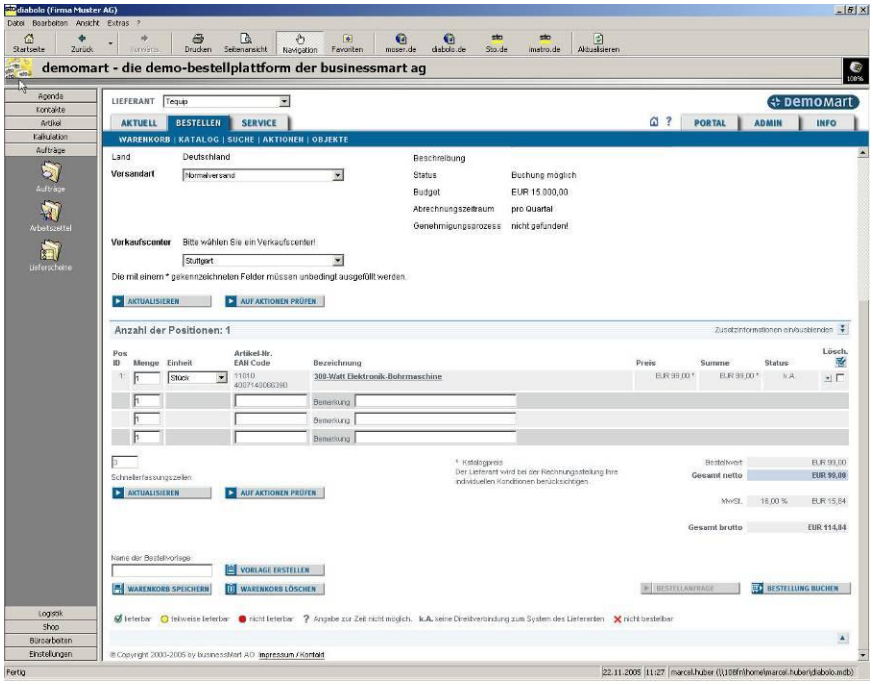

Figure 9 *PickBasket*

5.3. Fill the Shopping Basket

The shopping basket can be filled from the catalogue or the product search page. All suppliers can be selected. It is also possible to convert already stored order templates to new shopping baskets.

In general, all functionalities of the portal are available.

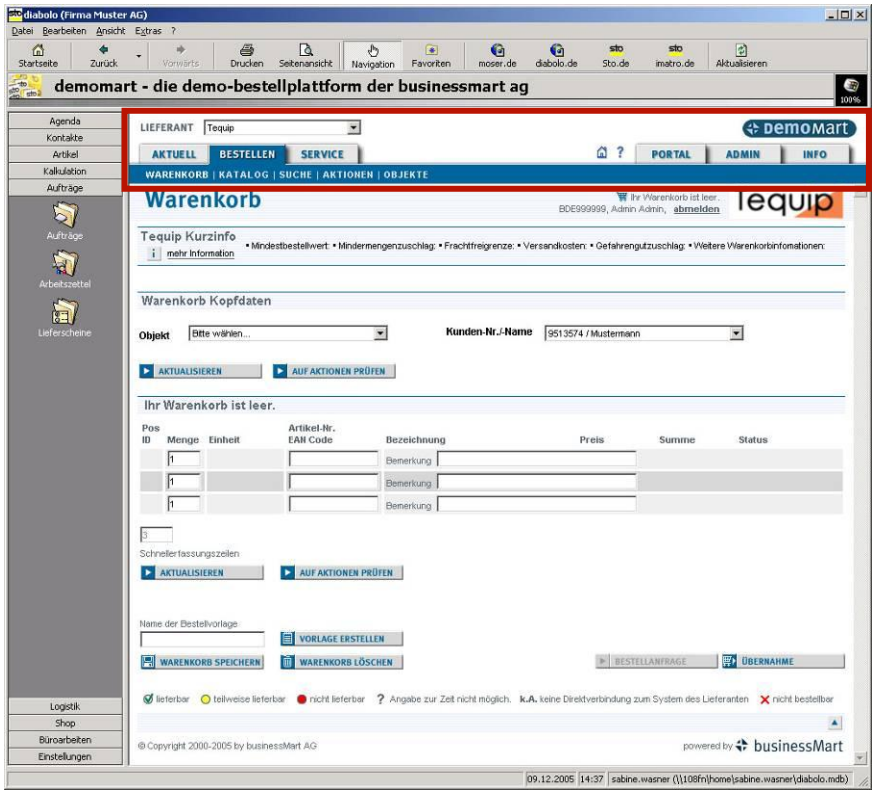

Figure 10 *All functionalities of the portal are available*

5.4. Takeover of Shopping Baskets

The shopping basket can be loaded into the own IT-System by clicking the “Apply” button.

The user can send an ERP request and check prices and availability of products before the shopping basket will be accepted. That is only possible when the supplier has an ERP connection.

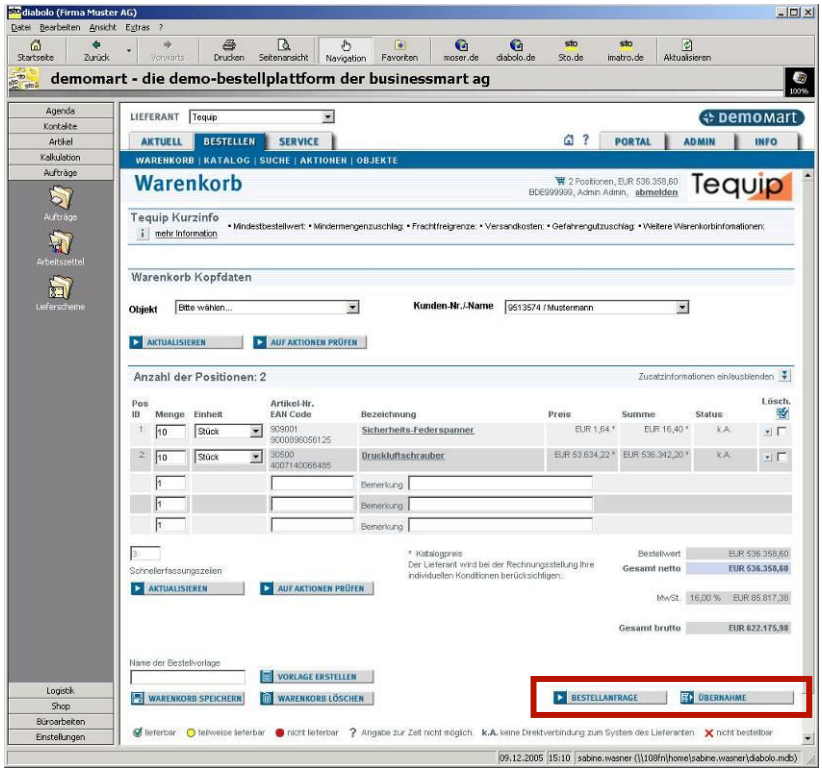

Figure 11 *ERP Request and Shopping Basket Takeover*

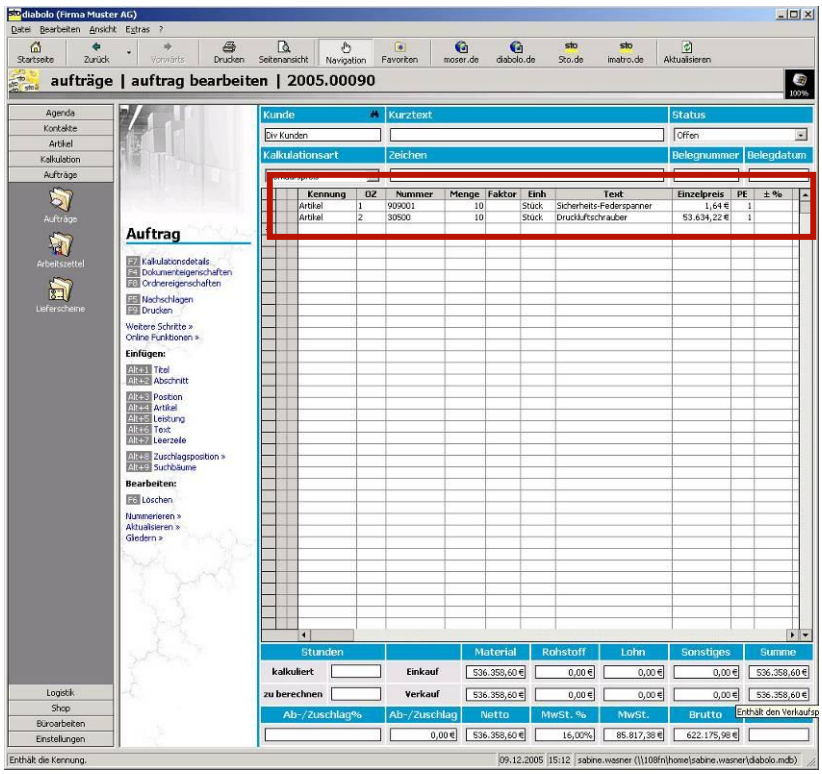

Figure 12 *Created Shopping Basket in the Users IT-System*

5.5. Already created Shopping Baskets

Already created shopping baskets can be accepted. In that case, the shopping baskets are saved in “Open Shopping Baskets”. The shopping baskets can be saved in the own IT-System by clicking “Place Order”.

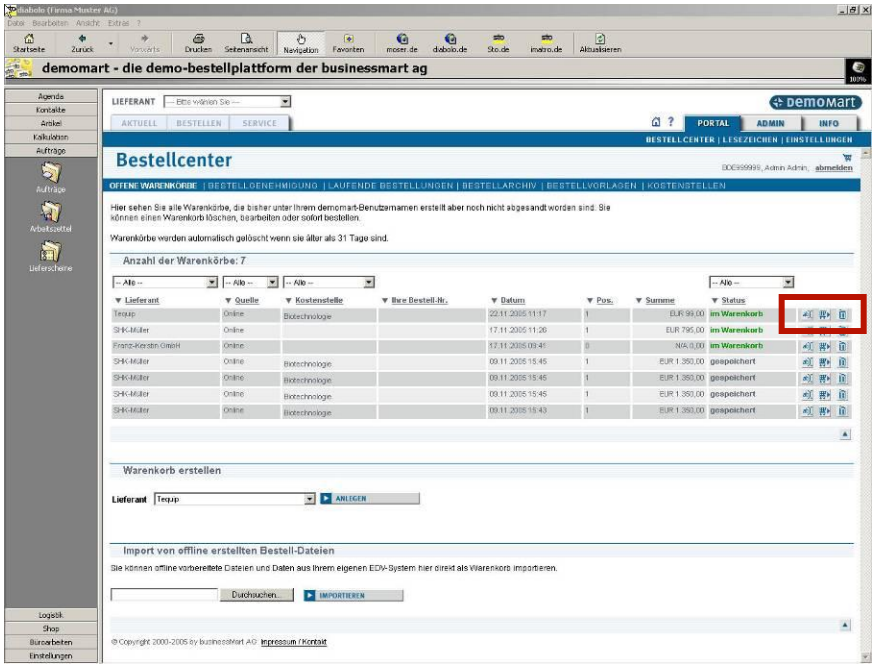

Figure 13 *Takeover from “Open Shopping Baskets”*

5.6. Technical Implementation

Devices of MartServices

MartLink PickBasket

The takeover of shopping baskets consists of a MartLink and a WebService. The MartLink PickBasket is used to access the portal from the own IT-System. The WebService GetBasket is used to pick up a shopping basket from the portal and to transfer the shopping basket into the IT-System of the user.

WebService GetBasket

After the takeover of the order, the MartLink returns supplier ID, shopping basket ID, checksum and TransActionNumber / expiration date.

5.7. MartLink PickBasket

5.7.1. URL

The MartLink-ID to PickBasket is “pickbasket”

https://‹Portal-URL›/martlink/ pickbasket

5.7.2. Request

The following parameters in the table apply for the MartLink OrderCenter, as well as all general parameters from chapter 3.

Name	Type	Required/ Optional	Description
TargetPage	String [“basket”, “catalog”, “productsearch“,“ openbaskets“]	required	This parameter is for choosing the entry point from shopping basket, catalogue, product search or open shopping baskets. The specification of a supplier (MartLink:SupplierId) is necessary for the entry into shopping basket, catalogue and product search

5.7.3. Response

The Response of the MartLink is initiated by clicking “Apply Shopping Basket”.

The following parameters in the table apply for the MartLink ProductDetail, as well as all general parameters from chapter 3.

Name	Type	Required/ Optional	Description
SupplierId	String [128] case insensitive	guaranteed	The SupplierId identifies the supplier of the compiled shopping basket.
BasketId	String [128] case sensitive	guaranteed	Identifier of the selected shopping basket. The ID is unique for the particular portal.

Name	Type	Required/ Optional	Description
CheckSum	Integer	not yet implemented	A checksum of the selected shopping basket will guarantee that no changes in the shopping basket remain unnoticed during the selection and the further process of the shopping basket.
TAN	Integer	not yet implemented	The transaction number for the WebService GetBasket is required and allows invalidating the process PickBasket.
ValidUntil	DateTime	not yet implemented	ValidUntil is the information about how long the Transaction ID stays validity.

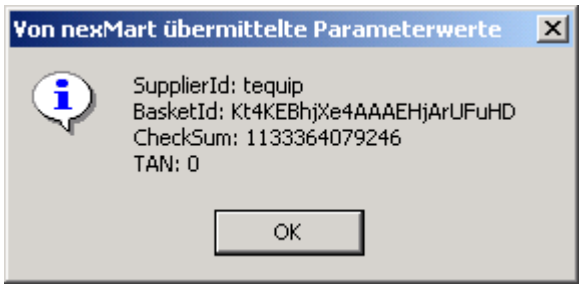

Figure 14 *Example for the unhidden Response*

5.8. WebService GetBasket

We are happy to provide an example to illustrate the project (soapUI).

WebService GetBasket

5.8.1. URL

URL for the Test-System:

http://ws-test.businessmart.de/services/emart/ExportBasket/2009-06-23

URL for the Live-System:

http://ws.businessmart.de/services/emart/ExportBasket/2009-06-23

The related Schema is available at http://api-test.businessmart.de/services/emartservices/doc/xsd_HtmlDoc/ExportBasketData.xsd.html.

The routing of the WebService is performed by the URL and the general WebService request parameter. The following are the relevant parameters with their current values:

» WebService:PortalID
» WebService:organization
» WebService:userName
» WebService:password
» WebService:basketId

5.8.2. Format of the Shopping Basket

The shopping basket returns as bM-OrderXML V1.1 (bmXML-1.11) in the answer of theWebService SOAP document. Ask for the specification of bM-OrderXML V1.1 in case you have not received these.

5.8.3. Exceptional Act

An exception is set for the WebService GetBasket:

» A Warning will be created if the shopping basket gets modified during the process of the MartLink PickBasket and the WebService GetBasket. This is examined by the Checksum.
» The shopping basket returns with a Warning if the ERP-System of the supplier cannot be accessed or a reconciliation error occurs.
» An Error is shown if the shopping basket cannot be exported.

» An Error is shown if the Basket:Id of a referenced shopping basket is not available on the referenced portal.

5.8.4. Request Parameter

The following parameters in the table apply for the WebService GetBasket, as well as all general parameters from chapter 4.

Name	Type	Required/ Optional	Description
basketId	String [128] case sensitive	required	Identifier of the selected shopping basket. The ID is unique for the particular portal.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns="http://api.businessmart.de/services/emartservices/exportTypes/2009-06-23">
  <soapenv:Header/>
  <soapenv:Body>
    <ns:getBasketRequest>
      <RequestHeader>
        <ClientId>SoapUI_Client</ClientId>
      </RequestHeader>
      <buyer>
        <portalId>abnahme.nexmart.de</portalId>
        <organization>BDE999999</organization>
        <userName>Vorname.Nachname</userName>
        <password>********</password>
      </buyer>
      <basketId>w5QKEBgjaRsAAAEmyPADUUu8</basketId>
    </ns:getBasketRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

Figure 15 *Example of a GetBasket-Request*

5.8.5. Response Parameter

The following parameters in the table apply for the WebService GetBasket, as well as all general parameters from chapter 4.

Name	Type	Required/ Optional	Description
basket	bmXML-1.11.xsd	optional	The shopping basket returns as Order/Basket XML. In case of failure skip this part.
deliverableState			available = requested amount is available partial_available = requested amount is partly available not_available = requested amount is not available not_requested = request was not check by the ERP-System (see command)

```
‹S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/›
  ‹S:Body›
    ‹ns2:getBasketResponse
xmlns:ns2="http://api.businessmart.de/services/emartservices/exportTypes/2009-06-23"›
      ‹ResponseHeader›
        ‹ResponseId›bbd10c317c8270e816b9e91ec2d6‹/ResponseId›
      ‹/ResponseHeader›
      ‹basket transactionId="undefined" formatVersion="bmXML-1.11"›
        ‹docs›
          ‹doc sequenceNumber="1"›
            ‹docHeader version="1.11" type="Basket"›
              ‹documentInfo source="eMartSilentCommerce"›
                ‹creationDate›2010-01-20T13:37:00Z‹/creationDate›
                ‹documentNumber›w5QKEBgjaRsAAAEmyPADUUu8‹/documentNumber›
```

```
    <status>NEW_EXPORTED</status>
    <moduleName>basket</moduleName>
  </documentInfo>
  <supplier>
    <technicalName>tequip</technicalName>
  </supplier>
  <orderer role="normal">
    <portalAccount>
      <marketplace>nexMart</marketplace>
      <organisation>BDE999999</organisation>
      <login>Vorname.Nachname</login>
      <country>DE</country>
      <customerNumber>123456</customerNumber>
    </portalAccount>
    <exchangeFormat>bmXML-1.11</exchangeFormat>
  </orderer>
  <invoiceAddress>
    <postalAddress>
      <name value="Admin" type="first"/>
      <name value="Admin" type="last"/>
      <name value="nexMart GmbH &amp;amp; Co. KG" type="company"/>
      <street>Gropiusplatz 10</street>
      <postcode>70563</postcode>
      <city>Stuttgart</city>
      <country>DE</country>
    </postalAddress>
  </invoiceAddress>
  <shipping type="normal">
    <shippingAddress/>
    <flagUseERPShipToAddress>true</flagUseERPShipToAddress>
  </shipping>
  <headerAttributes>
    <flagCompleteShipping>true</flagCompleteShipping>
    <currency>EUR</currency>
```

```
                <orderConfirmation value="jakob.sroda@businessmart.de" orderResponseFlag="false"
addressType="mail"/>
              </headerAttributes>
            </docHeader>
            <positions>
              <position deliverableState="not_requested" orderableState="orderable" id="1">
                <quantity value="10.0" unit="" type=""/>
                <article>
                  <manufacturerArticleNumber>60215961</manufacturerArticleNumber>
                  <articleName>Freie Position</articleName>
                  <catalogueArticleCode>60215961</catalogueArticleCode>
                  <catalogueNumber>metabo</catalogueNumber>
                  <articlePrice type="list">
                    <value>0.0</value>
                  </articlePrice>
                </article>
                <positionPrice type="list">
                  <totalValue>0.0</totalValue>
                </positionPrice>
                <positionAttributes>
                  <flagFreePosition>true</flagFreePosition>
                </positionAttributes>
                <results>
                  <result reference="//position/article/catalogueArticleCode" code="EMBPOS202030"
level="warning">
                    <customAttribute value="Free position " type="string" name="textMessage"/>
                  </result>
                </results>
              </position>
            </positions>
            <docFooter>
              <counter>1</counter>
              <grandTotals>
                <grandTotalNetPrice>0.0</grandTotalNetPrice>
                <grandTotalTax>0.0</grandTotalTax>
```

```
            <grandTotalGrossPrice>0.0</grandTotalGrossPrice>
          </grandTotals>
        </docFooter>
        <results>
          <result reference="CACreateResult" code="EMBWKF101010" level="success">
            <customAttribute value="CreateResult success " type="string" name="textMessage"/>
          </result>
        </results>
      </doc>
    </docs>
    <results>
      <result reference="//basketId" code="EMBWKF301240" level="error">
        <customAttribute value="Basket not synchronized successfully with ERP for export"
type="string" name="textMessage"/>
      </result>
    </results>
  </basket>
  <results>
    <result reference="CACreateResult" code="EMBWKF101010" level="success">
      <customAttribute value="CreateResult success " type="string" name="textMessage"/>
    </result>
  </results>
 </ns2:getBasketResponse>
</S:Body>
</S:Envelope>
```

Figure 16 *Example for a Response of a Shopping Basket*

5.8.6. Results

The possible result codes are available at http://api-test.businessmart.de/services/emartservices/doc/eMartServices_ErrorCodes.html

6. Order Centre

6.1. Function

The status of current orders may be viewed on the MartLink “Order Centre”. In addition, existing orders can be accessed via “Order Archive”.

6.2. Entry Point

The entry point into “Current Orders” or “Order Archive” can be set as default.

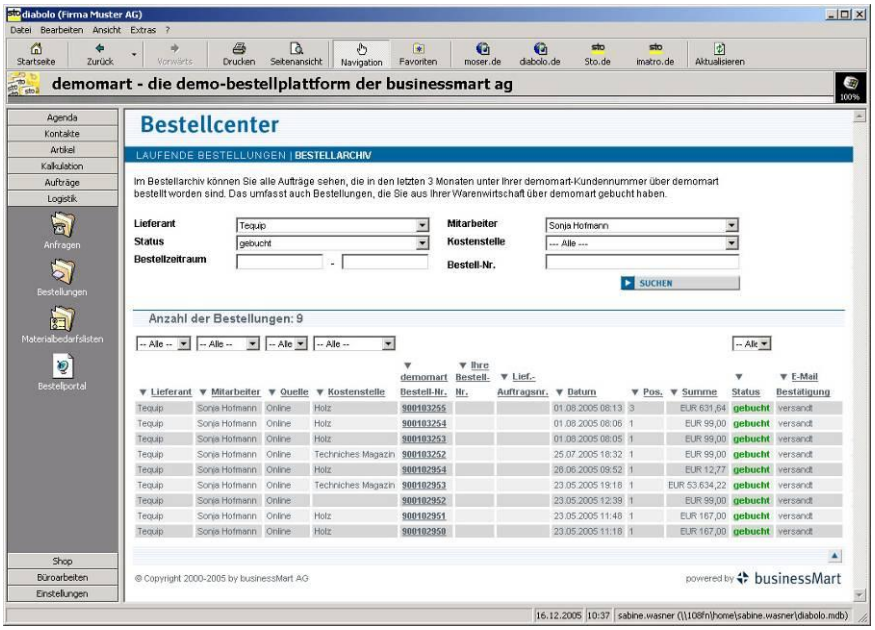

Figure 17 *Entry point into the Order Centre*

6.3. Search and Filter

Orders can be filtered and searched according to certain criteria. The user can select the filter or enter keywords on the interface. These settings can also be displayed automatically when the Order Centre is called. These settings can be specified as request parameters.

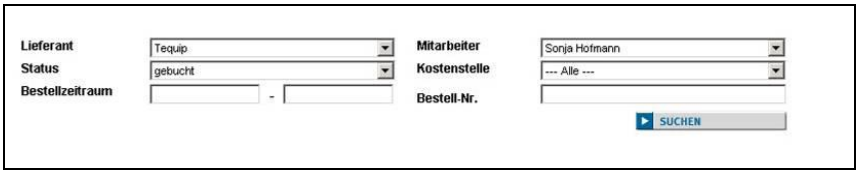

Figure 18 *Input and filtration to search for orders*

6.4. Differences to the Online Features

The MartLink cannot perform all functions of the storefront. This applies mainly to the following functions:

In the detailed view of the orders are no actions available, e.g. takeover of positions into the shopping basket.

It is not possible to cancel an order during the booking process or generate a shopping basket out of order data if an error occurs in “Current Orders”.

6.5. Technical Implementation

6.5.1. URL

The MartLink-ID for PickBasket is “ordercenter”.

https://<Portal-URL>/martlink/ordercenter

6.5.2. Request

The following parameters in the table apply for the MartLink OrderCenter, as well as all general parameters from chapter 3.

Name	Type	Required/ Optional	Description
TargetPage	String [“openorders”, “archive”]	Required	Choice of entry point among “Current Orders” or “Order Archive”. In case of missing or invalid parameter the entry point is “Order Archive”. Note: If “Open Orders” is selected you get redirected to “Current Orders”. On this page are no search parameters. That means, are such parameters set, they will be ignored.
OrderNo	String [255] case sensitive	Optional	The optional OrderNo leads to the call of the appropriate order. The OrderNo is the eMart order number (OrderId) or the order number of the customer. Is there no order to the OrderNo it will be rejected.
FromDate	Date YYYY-MM-DD (ISO 8601)	Optional	Used to filter the orders by date.

Name	Type	Required/ Optional	Description
ToDate	Date YYYY-MM-DD (ISO 8601)	Optional	Used to filter the orders by date.
OrderStatus	Num	Optional	Used to filter the orders by status. -1 booked 1000 booked 1200 not booked/incorrect 500 in progress 0 unknown
BuyerName	String [255] case sensitive	Optional	Used to filter the orders by person. The persons have a login on the portal. The personal user name consists mostly ‹first name›.‹last name›.
CostCenterID	String [255]	Optional	Used to filter the orders by cost centers. This only makes sense if the user is using E-Procurement II.
ExecSearch	String [„true“, „false”]	Required	The parameter sets the automatic start of search.

6.5.3. Response

The MartLink OrderCenter has no Response parameter.

All general parameters of chapter 3 apply to the MartLink OrderCenter.

7. Order Tracking

7.1. Function

The MartLink ERPOrderHistory calls the function “Order Overview” of the portal. In this case, all orders of the selected supplier will be displayed. This information is obtained directly from the ERP-System of the supplier.

Thus obtains all important information of an order:

» Current status of an order (booked, delivered, partly delivered, etc.)
» All information about the order itself (price, position, etc.)

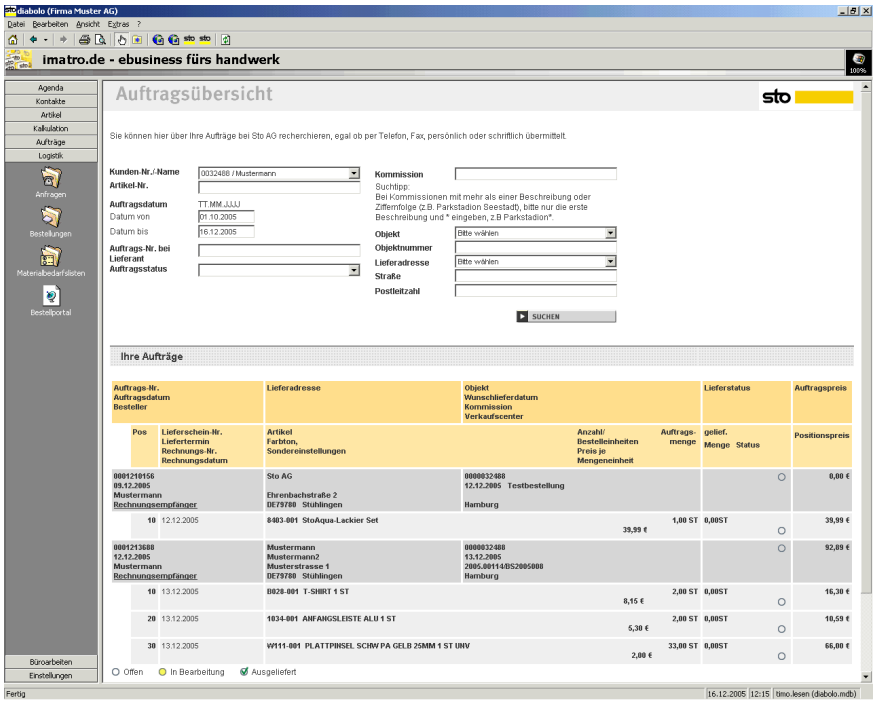
diabolo (Firma Muster AG)
imatro.de - ebusiness fürs handwerk
Auftragsübersicht
Sie können hier über Ihre Aufträge bei Sto AG recherchieren, egal ob per Telefon, Fax, persönlich oder schriftlich übermittelt.
Ihre Aufträge

Figure 19 *Order Tracking*

7.2. Search and Filter

Orders and information can be searched or filtered by the following criteria:

- Customer order number or name
- Order number of the supplier
- Article number
- Order date from/to
- Order number from the supplier
- Order Status

Manual Input

The function can be implemented in a way that the page is called without default search and filter criteria.

Default Input

It is also possible that search and filter criteria are already set. The results of the criteria are displayed immediately without the users input on the portal interface.

Automatic Search

Furthermore, it can be decided whether the search is to be started automatically or initiated by the manual click of the user.

7.3. Technical Implementation

7.3.1. URL

The MartLink-ID for ERPOrderHistory is “orderhistory”

https://‹Portal-URL›/martlink/orderhistory

7.3.2. Request

The following parameters in the table apply for the MartLink OrderCenter, as well as all general parameters from chapter 3.

Name	Type	Required/ Optional	Description
SupplierID	String [0..128] case insensitive	Required	A valid identifier of a supplier who is part of the portal. Default is the preset preferred supplier on the portal.
ExecSearch	String [„true“, „false”]	Optional	The parameter sets the automatic start of search.
OrderState	Num	Optional	This setting allows filtering orders according to their process status. 0 All 1 In process at suppliers 2 Partly delivered 3 Delivered
FromDate	String YYYY-MM-DD (ISO 8601)	Optional	Used to filter orders by date.
OrderState	String YYYY-MM-DD (ISO 8601)	Optional	Used to filter orders by process status.
BuyerOrderNo	String [255]	Optional	The order number of the customer.
SellerOrderNo	String [255]	Optional	The order number of the supplier.

7.3.3. Response

The MartLink ERPOrderHistory has no Response parameter. All general parameters of chapter 3 apply to the MartLink OrderCenter.

8. Product Configuration

8.1. Function

It is possible to configure and order products with certain features, such as sizes for screwdrivers or hue for paints, on all eMart portals.

A setting page for the product configuration is available by means of the GTIN (EAN) code or order number and supplier name. Optionally, an existing configuration of a product can be passed to the MartLink. This is used as preselection for the relevant product attributes.

8.2. Configuration of Products

The configuration may differ from product to product and depends on the settings of the seller.

Especially the following configurations are possible:

» Product colors
» Advanced setting – Extended Warranty
» Advanced setting – Purpose wet/color/fabric (e.g. sandpaper)

8.3. Product Colors

Product colors can be configured for some products such as lacquers. In the ERP-program of the customer opens a window “Advanced Settings” when the MartLink ConfigureProduct is called. The customer can select the relevant color in the opened window (see figure 20).

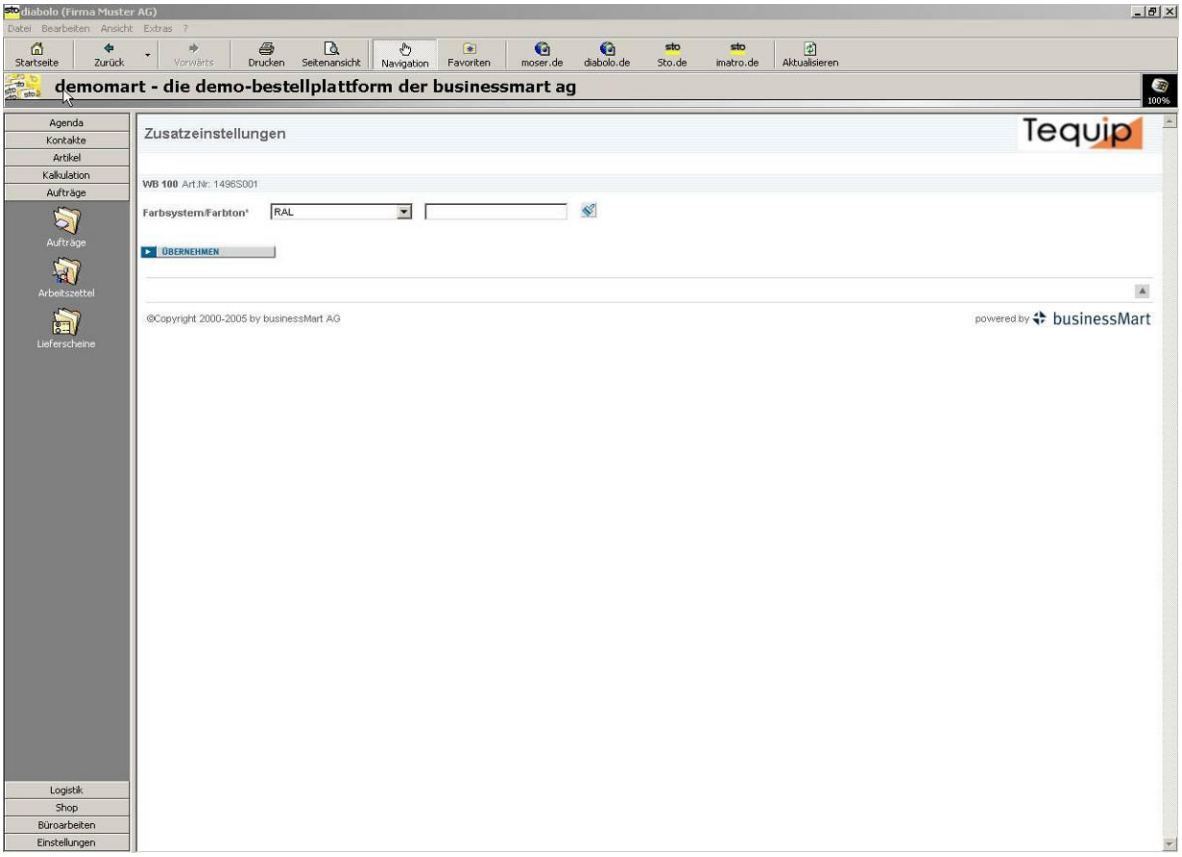

Figure 20 *Configuration of Product Colors*

The color test will open if the user clicks on the icon . All appropriate colors are listed after entering of the color system and the desired hue. The color of the product is configured after clicking the “Apply” button.

demomart - die Demo-Bestellplattform der businessMart AG - Micro...
Farbtonprüfung
Farbsystem*
RAL
Farbton*
PRÜFEN
Ihre Auswahl ergab 50 Farbtöne.
Farbsystem
Farbton
Sto-Farbton
RAL 1001 - Beige RAL1001 ÜBERNEHMEN
RAL 1002 - Sandgelb RAL1002 ÜBERNEHMEN
RAL 1004 - Goldgelb RAL1004 ÜBERNEHMEN
RAL 1006 - Maisgelb RAL1006 ÜBERNEHMEN
RAL 1018 - Zinkgelb RAL1018 ÜBERNEHMEN
RAL 1020 - Olivgelb RAL1020 ÜBERNEHMEN
RAL 1021 - Rapsgelb RAL1021 ÜBERNEHMEN
RAL 1036 - Perlgold RAL1036 ÜBERNEHMEN
RAL 3000 - Feuerrot RAL3000 ÜBERNEHMEN
RAL 3003 - Rubinrot RAL3003 ÜBERNEHMEN
RAL 3005 - Weinrot RAL3005 ÜBERNEHMEN
RAL 3009 - Oxidrot RAL3009 ÜBERNEHMEN
RAL 3011 - Braunrot RAL3011 ÜBERNEHMEN
RAL 3012 - Beigerot RAL3012 ÜBERNEHMEN
RAL 3014 - Altrosa RAL3014 ÜBERNEHMEN
Farbtöne 1 bis 15 von 50
1 2 3 4
ABBRECHEN
Fertig
Vertrauenswürdige Sites

Figure 21 *Color check*

The user gets a note if the required color does not exist (figure 21).

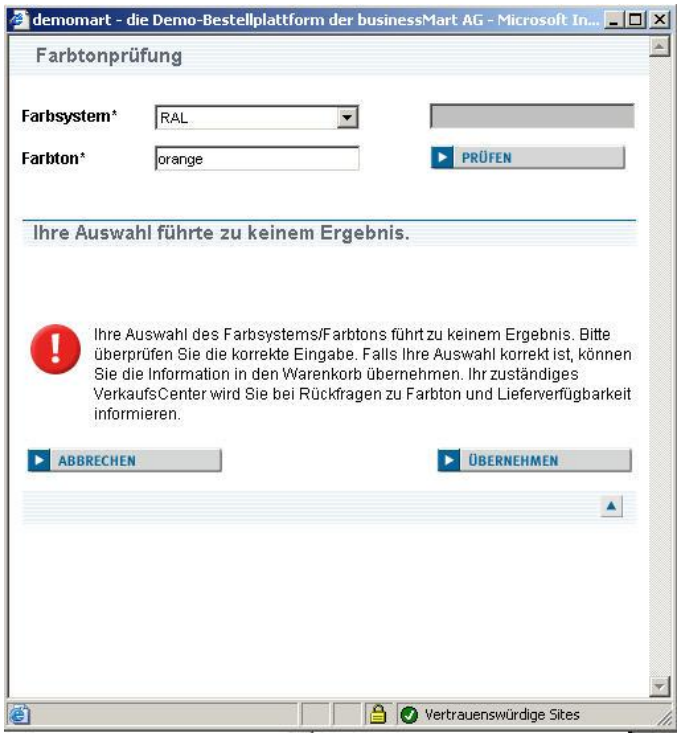
demomart - die Demo-Bestellplattform der businessMart AG - Microsoft In...
Farbtonprüfung
Farbsystem*
RAL
Farbton*
orange
PRÜFEN
Ihre Auswahl führte zu keinem Ergebnis.
Ihre Auswahl des Farbsystems/Farbtons führt zu keinem Ergebnis. Bitte überprüfen Sie die korrekte Eingabe. Falls Ihre Auswahl korrekt ist, können Sie die Information in den Warenkorb übernehmen. Ihr zuständiges VerkaufsCenter wird Sie bei Rückfragen zu Farbton und Lieferverfügbarkeit informieren.
ABBRECHEN
ÜBERNEHMEN
Vertrauenswürdige Sites

Figure 22 *Color does not exist*

The user gets a note if no product configuration is set (figure 22).

8.4. Technical Implementation

8.4.1. URL

The MartLink-ID for ProductDetail is “configureproduct”

Default values can be transferred for the product configuration in the request. These are set as a preselection. They will be ignored if at least one is invalid.

8.4.2. Request

The following parameters in the table apply for the MartLink ConfigureProduct, as well as all general parameters from chapter 3.

Name	Type	Required/ Optional	Description
SupplierArticle Number	String [1..128] case sensitive	required	Unique article number of the product given by the supplier. Mostly it is the order number or the GTIN (former EAN) Code.
conditioning1	String [0..255]	optional	The default settings for the product configuration can be passed in the fields conditioning1 to 5, colorNo and colorText.
conditioning2	String [0..255]	optional	
conditioning3	String [0..255]	optional	
conditioning4	String [0..255]	optional	
conditioning5	String [0..255]	optional	
colorNo	String [0..255]	optional	
colorText	String [0..255]	optional	

Table 4 *Request ConfigureProduct*

8.4.3. Response

The following parameters in the table apply for the MartLink ConfigureProduct, as well as all general parameters from chapter 3.

Name	Type	Required/ Optional	Description
conditioning1	String [0..255]	optional	The default settings for the product configuration will be passed in the field conditioning1 to 5.
conditioning2	String [0..255]	optional	
conditioning3	String [0..255]	optional	
conditioning4	String [0..255]	optional	
conditioning5	String [0..255]	optional	
colorNo	String [0..255]	optional	The parameter transmits the color code. The absolute term “color” is transmitted if no valid hue has been selected.
colorText	String [0..255]	optional	The user description of the color is inserted here if no valid hue has been selected in the parameter colorNo.

Table 5 *Response ConfigureProduct*

8.5. Advanced Settings

Another product configuration is the Extended Warranty (figure 23) and Purpose (figure 24).

diabolo (Firma Muster AG)
demomart - die demo-bestellplattform der businessmart ag
Zusatzeinstellungen
Tequip
Agenda
Kontakte
Artikel
Kalkulation
Aufträge
300-Watt Elektronik-Bohrmaschine Art.Nr: 7777-001
Garantieverlängerung
3 Jahre
ÜBERNEHMEN
Ihre Auswahl des Farbsystems/Farbtons führt zu keinem Ergebnis. Bitte überprüfen Sie die korrekte Eingabe. Falls Ihre Auswahl korrekt ist, können Sie die Information in den Warenkorb übernehmen. Ihr zuständiges VerkaufsCenter wird Sie bei Rückfragen zu Farbton und Lieferverfügbarkeit informieren.
©Copyright 2000-2005 by businessMart AG
powered by businessMart

Figure 23 *Advanced Settings - Extended Warranty*

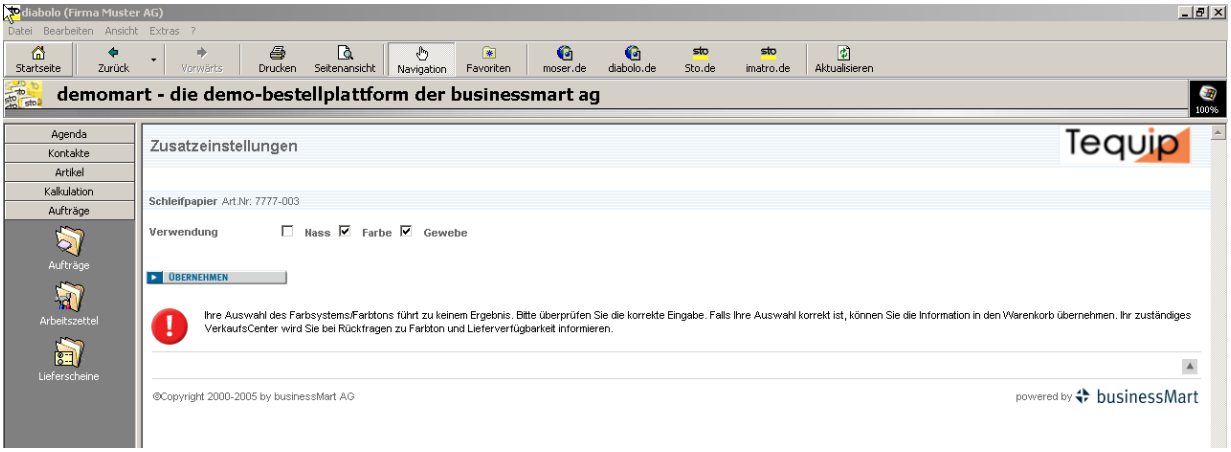
diabolo (Firma Muster AG)
demomart - die demo-bestellplattform der businessmart ag
Zusatzeinstellungen
Tequip
Agenda
Kontakte
Artikel
Kalkulation
Aufträge
Schleifpapier Art.Nr: 7777-003
Verwendung
Nass
Farbe
Gewebe
ÜBERNEHMEN
Ihre Auswahl des Farbsystems/Farbtons führt zu keinem Ergebnis. Bitte überprüfen Sie die korrekte Eingabe. Falls Ihre Auswahl korrekt ist, können Sie die Information in den Warenkorb übernehmen. Ihr zuständiges VerkaufsCenter wird Sie bei Rückfragen zu Farbton und Lieferverfügbarkeit informieren.
©Copyright 2000-2005 by businessMart AG
powered by businessMart

Figure 24 *Advanced Settings - Purpose*

9. MartLink AutoLogin

The MartLink AutoLogin is a call of the eMart surface with an automatic login.

Full Functionality

The surface contains the full functionality of eMart with all menus etc.

Regular CI

The surface has to appear in the regular portal/shop CI.

Direct entry to Shopping Basket

The user goes direct to the shopping basket by calling the parameter BasketID.

Verification of Rights

The rights of the user login and the authority of the requested function are checked.

Fallback

The login is based on the settings of the user if no BasketID is handed over by the system.

Call of URL

https://‹Portal-URL›/martlink/autologin

Call of Parameter

Parameter	Required	OrderScan configuration value
BuyerOrganization	Yes	BuyerOrganization
BuyerUserName	Yes	BuyerUserName
BuyerUserPassword	Yes	BuyerUserPassword
BasketID	No	

Parameter	Required	OrderScan configuration value
MartLinkLocale	No	Locale
MartLinkCurrency	No	Currency
ClientTargetURL	Yes	ClientTargetURL
ClientName	Yes	ClientName
ClientRequestMethod	Yes	ClientRequestMethod
MartLinkId	Yes	MartLinkId
MartLinkVersion	Yes	MartLinkVersion
MartLinkPortalID	Yes	PortalID

Error Handling

NoRightStorefrontMainLogin – No rights for login

Your account has not yet been released!

Your login is failed.

You can start to work if you are registered. You have to enter your account number (BDE...), name and password. Please contact your in-house manager for this information.

FirstLogin - Login has to be done first

Your data have to be confirmed!

Your login is failed.

You can start to work if you finished the registration process on the homepage of the portal. Please enter your account number (BDE...), name and password on the homepage and confirm your data. Please contact your in-house manager for this information.

NoBasketViewRight – No rights to view the shopping basket

The requested action could not be executed because the necessary right is not activated. Rights are managed by your in-house manager.

Please log off and on again if you were able to perform this operation before. Please contact your in-house manager if you receive this message again.

10. Prices and Availability

10.1. Function

Current prices and availabilities of products can be checked by the MartService order request. The order request is done with a filled shopping basket, which is enriched with the latest information from the ERP system of the supplier.

There are three different courses of action to start an order request with a shopping basket. All three courses of action are possible with certain components of the MartServices.

10.2. Prices und Availabilities in the own Shopping Basket

The Process

The client creates a shopping basket in his own computer and requests prices and availabilities of products by clicking a button like “Order Inquiry”. As a result, the latest data are displayed in the shopping basket.

10.2.1. Practicable Example with Screenshots

The processes are not perceived by the user. It is just one click to get the latest prices and availabilities of products. In this case, the technical modules are linked in a way that they start each other.

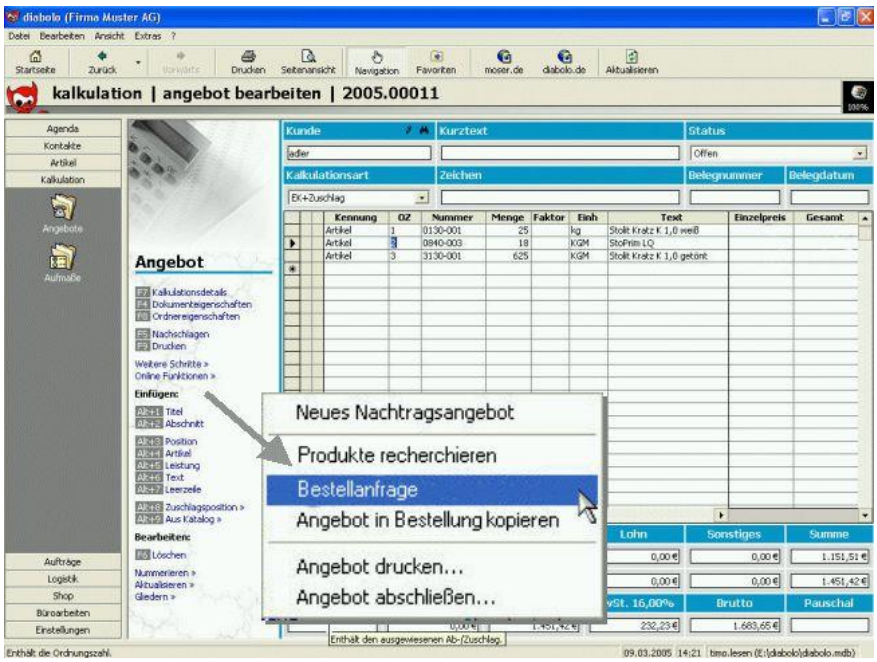

Figure 25 *Possible Implementation “Order Request“*

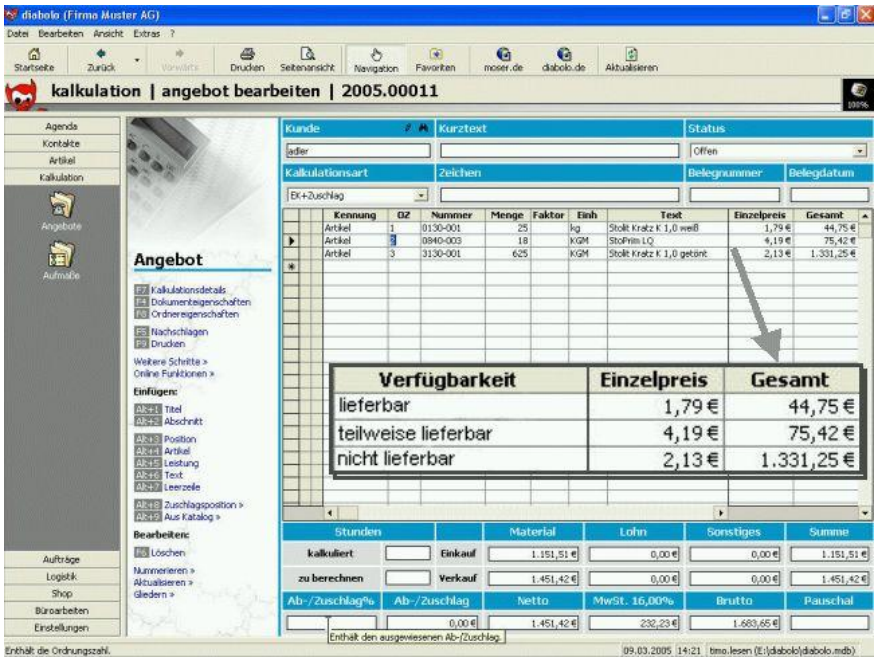

Figure 26 *Prices and Availabilities of Products*

10.2.2. Attributes/Process

MartService Modules

WebService
Get Basket

The following happens in the background: The shopping basket will be sent to the portal to start an order request from the ERP system of the supplier. In order to use the feature price and availability of products a ticket number returns to the computer system during the upload of the shopping basket. That allows the user to ask for prices and availabilities of products. The order request is started automatically on the portal after the shopping basket is arrived. The computer system retrieves the requested data of the shopping basket by the ticket number.

The following is a representation of the scenario described above:

1. The user starts to upload the shopping basket as XML to the portal by a click or something similar (depends on the implementation). The shopping basket is visible in the portal under “Open Basket”.
2. The user’s computer system gets a ticket number back, which identifies the submitted shopping basket.
3. At the same time, an order request is automatically started. The shopping basket is requested in the ERP system of the supplier by the portal.
4. The current data return and the shopping basket is ready for collection in the OrderCentre of the portal.

5. The EDP system makes a request to the portal to collect the shopping basket (SB) by using the ticket number (TN). It is recommended to wait 5 seconds plus 2 seconds for each position before a renewed pick up of the shopping basket is possible.
6. The shopping basket will be sent in XML format to the computer. The state of the shopping basket can be determined from the Resultcode.

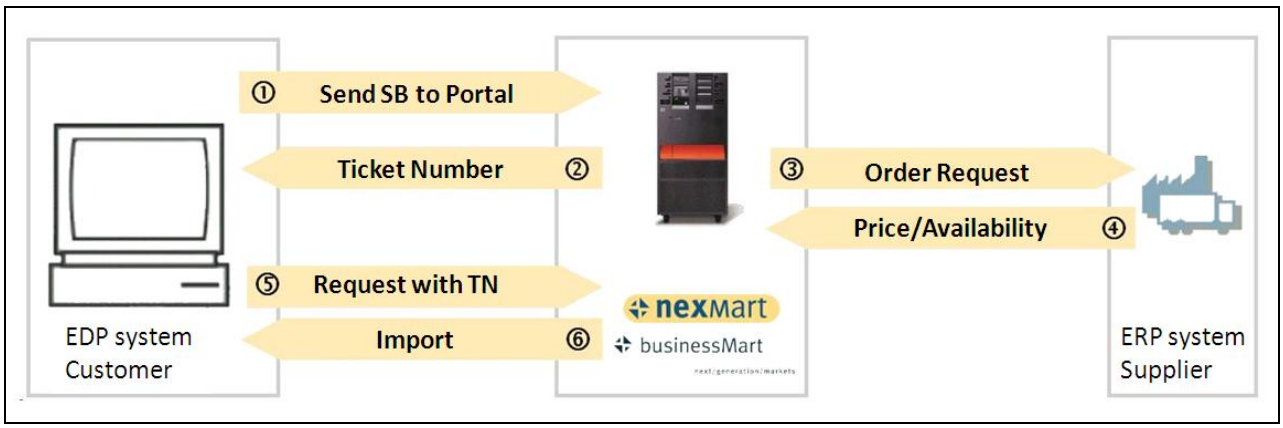

Figure 27 *Scenario 1: Schematic Process*

10.2.3. Technical Implementation

The scenario is technically implemented with WebServices. You will find specific information in this chapter. Please read also the general information about WebServices from chapter 11.

The following modules are needed for scenario 1:

» WebService UploadBasket
» WebService GetBasket

WebService UploadBasket

URL for the Test-System:

http://ws-test.nexmart.net/services/emart/UploadBasket/2010-08-17

URL for the Live-System:

http://ws.nexmart.net/services/emart/UploadBasket/2010-08-17

The related schema is available at http://ws.nexmart.net/services/emart/UploadBasket/2010-08-17?wsdl.

We are happy to provide you an example of this project (soapUI).

Shopping Basket-ID

The Basket-ID is used to identify the shopping basket on the corresponding portal. The Basket UUID of the shopping basket is used to map this ID.

Parameter

The following parameters in the table apply for the WebService GetBasket, as well as all general parameters from chapter 4.

Name	Type	Required/ Optional	Description
command	String	Optional	An ERP request is made if “simulate” is transferred.
basket	bmXML-1.11.xsd	Required	The shopping basket as Order/Basket XML document.

Table 6 *Request Parameter WebService Upload Basket*

Name	Type	Required/ Optional	Description
basketId	String	Required	Identifier of the selected shopping basket. The ID is unique for the particular portal.

Table 7 *Response Parameter WebService UploadBasket*

The possible Result-Codes are available at http://api-test.businessmart.de/services/emartservices/doc/eMartServices_ErrorCodes.html.

WebService GetBasket

See chapter „WebService GetBasket”

Name	Type	Required/ Optional	Description
basketId	String [128] case sensitive	Required	Identifier of the selected shopping basket. The ID is unique for the particular portal.

Table 8 *Request-Parameter for GetBasket*

```
‹soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns="http://api.businessmart.de/services/emartservices/exportTypes/2009-06-23"›
  ‹soapenv:Header/›
  ‹soapenv:Body›
    ‹ns:getBasketRequest›
      ‹RequestHeader›
        ‹ClientId›SoapUI_Client‹/ClientId›
      ‹/RequestHeader›
      ‹buyer›
        ‹portalId›abnahme.nexmart.de‹/portalId›
        ‹organization›BDE999999‹/organization›
        ‹userName›Vorname.Nachname‹/userName›
        ‹password›********‹/password›
      ‹/buyer›
      ‹basketId›w5QKEBgjaRsAAAEmyPADUUu8‹/basketId›
    ‹/ns:getBasketRequest›
  ‹/soapenv:Body›
‹/soapenv:Envelope›
```

Figure 28 *Example of a Shopping Basket Request*

Link of Modules

The two modules have to be linked together so that the user can call prices and availability with one click.

The order request is automatically triggered once the shopping basket was sent on the portal by UploadBasket. This function is controlled by the parameter “command” with the value “simulate“.

The order request is sent to the ERP system. Then the request is edited and returns. After that, the prices and availability are ready for retrieval. The request to collect the shopping basket has to be sent automatically by the computer after a certain time. It is recommended to wait 5 seconds plus 2 seconds for each position before a renewed pick up of the shopping basket is possible. The collection may be performed several times until the desired state of the shopping basket is reached. The shopping basket cannot be collected again if the synchronization is completed.

10.3. View Prices and Availability

The Process

Again, the client creates a shopping basket in his own computer and requests prices and availabilities of products by clicking a button. Then a pop-up with the information will be called. The shopping basket cannot be changed that way and is only used for viewing.

10.3.1. Practicable Example with Screenshots

The processes are not perceived by the user. It is just one click to get the latest prices and availabilities of products. In this case, the technical modules are linked in a way that they start each other.

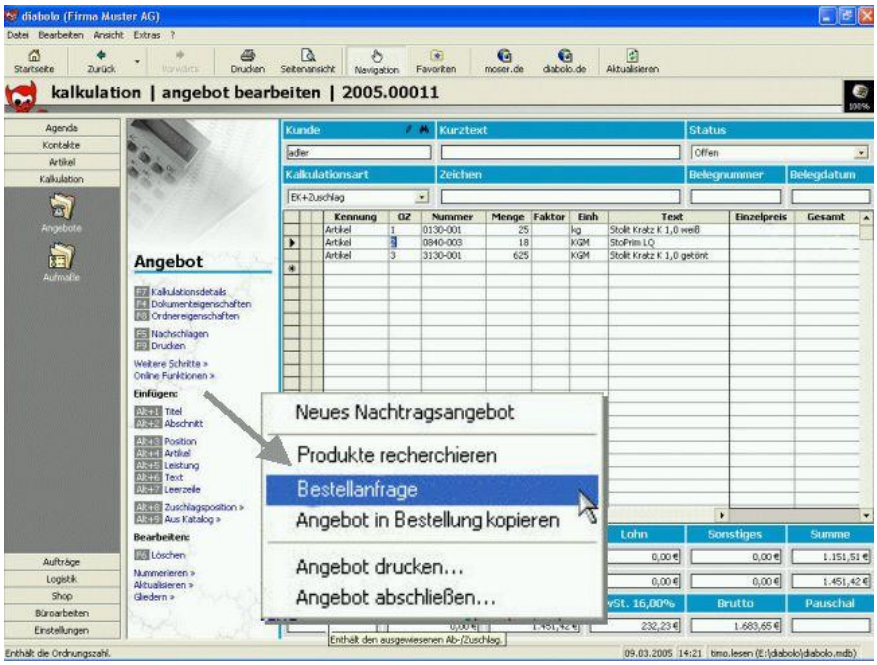

Figure 29 *Possible Implementation "Order Request"*

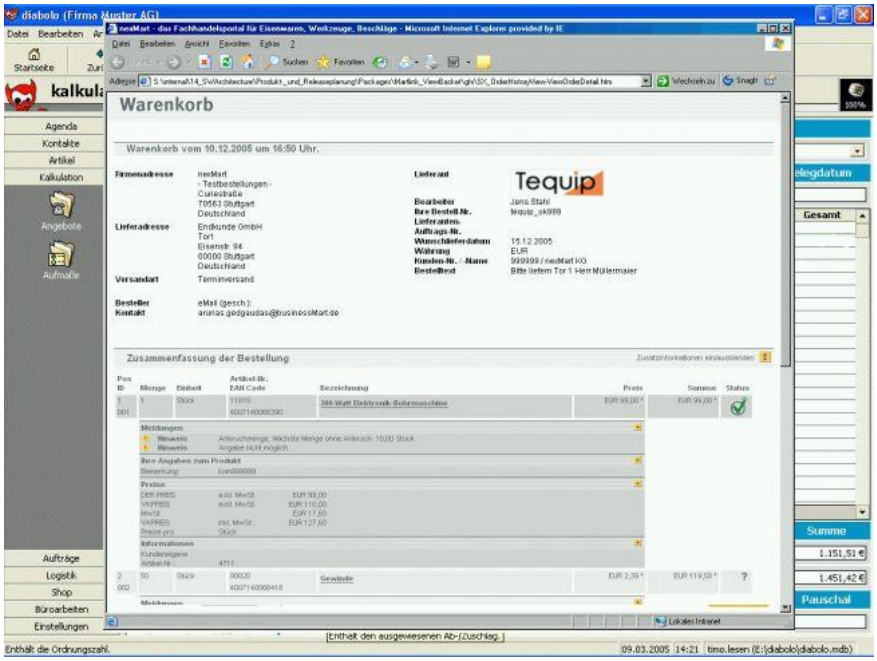

Figure 30 *Prices and Availability shown via Pop-Up*

10.3.2. Attributes/Process

MartService Modules

WebService UploadBasket

MartLink ViewBasket

The following happens in the background: The shopping basket will be sent to the portal to start an order request from the ERP system of the supplier. In order for the user request, a ticket number returns to the computer system during the upload of the shopping basket. That allows the user to ask for prices and availabilities of products. The order request is started automatically on the portal after the shopping basket is arrived.

The computer system can send a request to the portal by using the ticket number. In that way, the basket will appear as a pop-up on the screen of the customer.

All users of a BDE can call the shopping baskets of their colleagues because the shopping baskets are handled by the Order Centre.

The following is a representation of the scenario described above:

1. The user starts to upload the shopping basket as XML to the portal by a click or something similar (depends on the implementation). The shopping basket is visible in the portal under “Open Basket”.
2. The user’s computer system gets a ticket number back, which identifies the submitted shopping basket.

3. At the same time, an order request is automatically started. The shopping basket is requested in the ERP system of the supplier by the portal.
4. The current data return and the shopping basket is ready for collection in the OrderCentre of the portal.
5. The EDP system makes a request to the portal for displaying the shopping basket (SB) by using the ticket number (TN).
6. The shopping basket will be displayed as a pop-up.

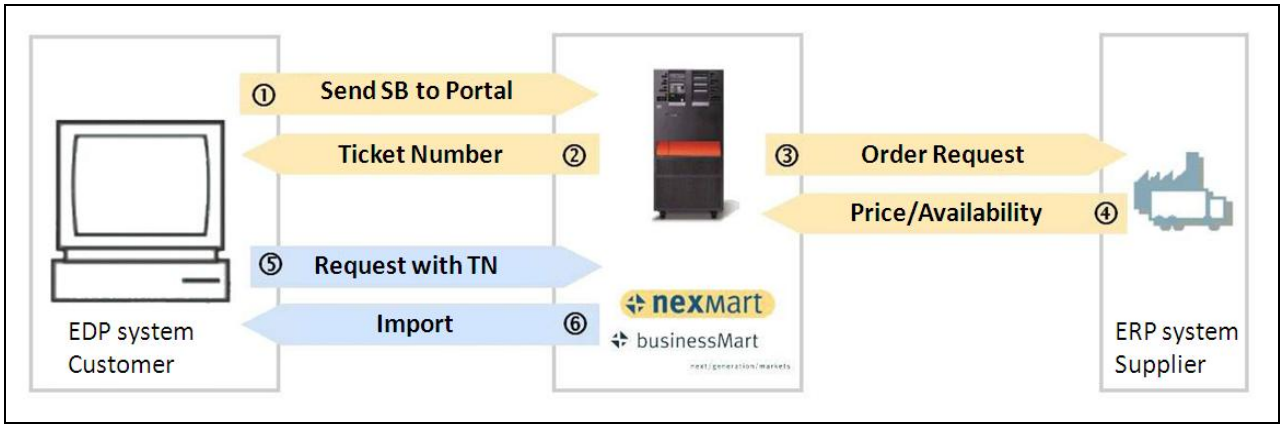

Figure 31 *Scenario 2: Schematic Process*

Attributes of Shopping Basket View

- The interface is displayed in the respective colors of the portal.
- Basis of representation is OrderDetail from the MartLink OrderCenter.
- The title of the page is “Shopping Basket“, the first header: “Shopping Basket of [dd.mm.yyyy] at [hh:mm] am/pm“
- All buttons are removed. The only possible action is to open and collapse the position fields.
- Characteristics of the modules fieldservice and pricing are not included in this package.

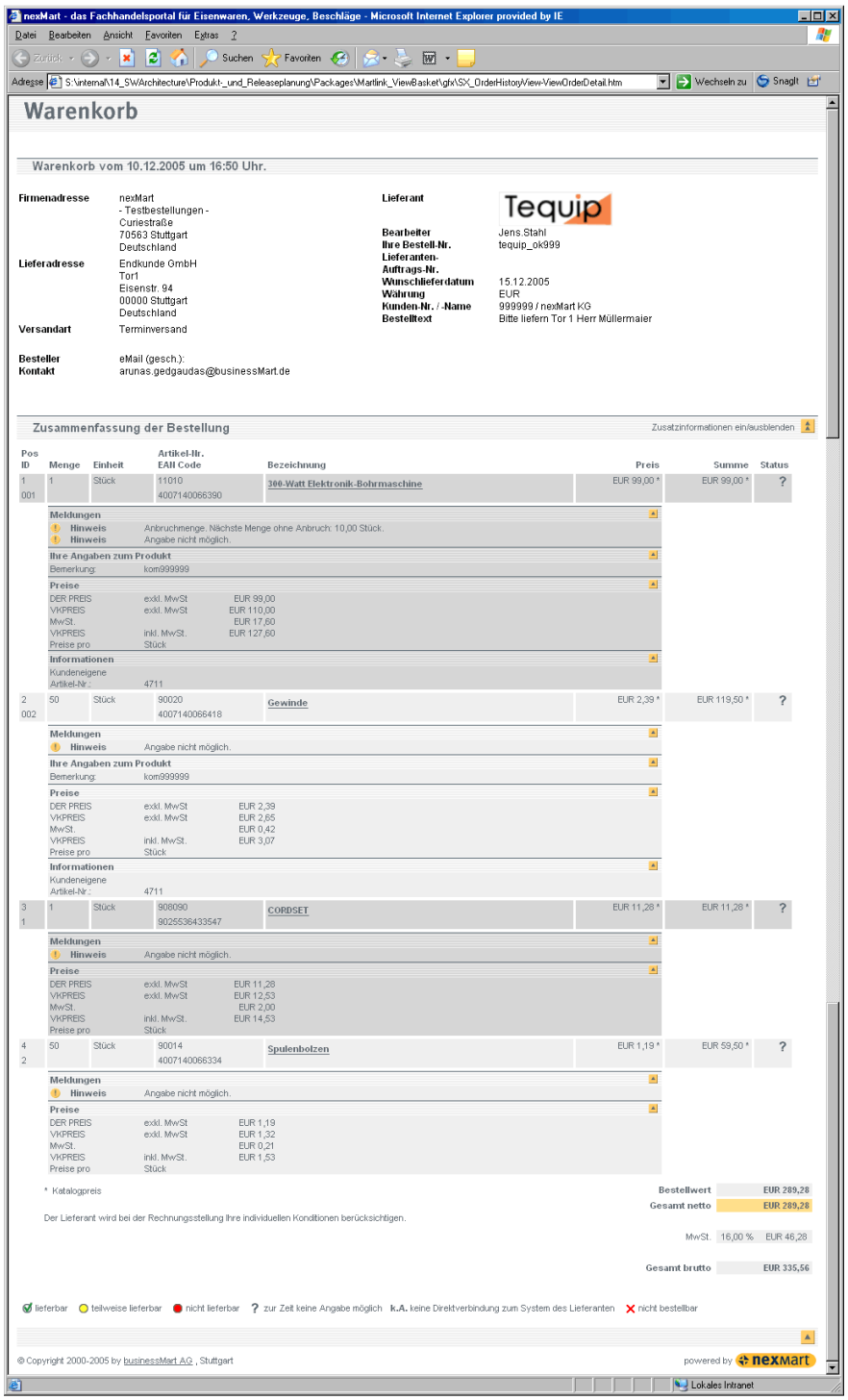

Figure 32 *Shopping Basket shown via Pop-Up*

Case of Failure

The sent shopping baskets cannot be corrected in the SCCC. It is possible to correct them by using the Order Center of the portal or by repeating the transmission of the shopping basket from the computer system. Especially a repetition of the transmission makes sense when the Martservice GetBasket is implemented in the customers ERP.

A further correction can happen to the SCCC customer order correction.

ERP Response takes longer

It takes a view seconds until the shopping basket is transmitted to the portal and the order request is edited by the ERP system.

A demand note to be patient is displayed when the pop-up of the shopping basket is called before the process of editing is done.

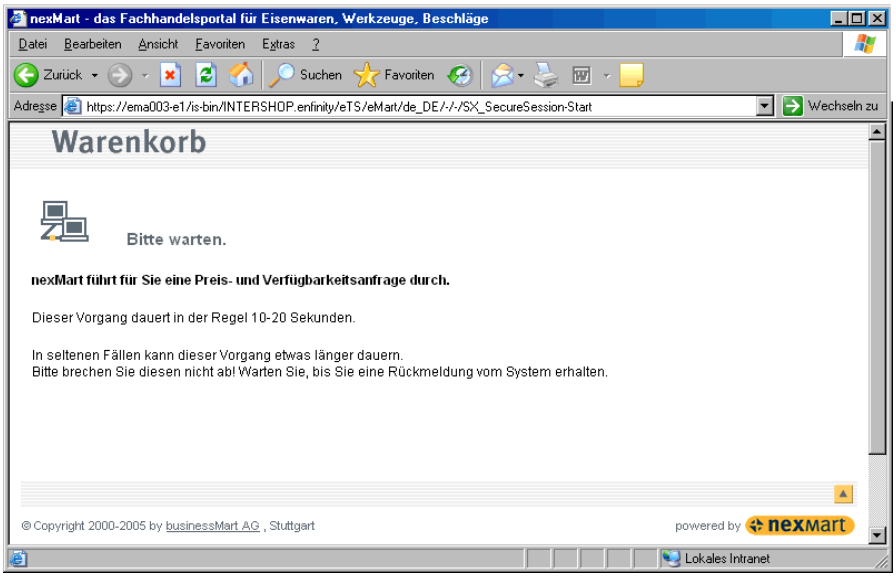
nexMart - das Fachhandelsportal für Eisenwaren, Werkzeuge, Beschläge
Datei Bearbeiten Ansicht Favoriten Extras ?
Zurück Suchen Favoriten
Adresse https://ema003-e1/is-bin/INTERSHOP.enfinity/eTS/eMart/de_DE/-/-/SX_SecureSession-Start
Wechseln zu
Warenkorb
Bitte warten.
nexMart führt für Sie eine Preis- und Verfügbarkeitsanfrage durch.
Dieser Vorgang dauert in der Regel 10-20 Sekunden.
In seltenen Fällen kann dieser Vorgang etwas länger dauern.
Bitte brechen Sie diesen nicht ab! Warten Sie, bis Sie eine Rückmeldung vom System erhalten.
© Copyright 2000-2005 by businessMart AG , Stuttgart
powered by nexMart
Lokales Intranet

Figure 33 *Waiting for the Order Request Page*

10.3.3. Technical Implementation

The scenario is technically implemented with a WebServices and a MartLink. You will find specific information in this chapter. Please read also the general information about MartLinks and WebServices from chapter 3 and 4.

The following modules are needed for scenario 2:

» WebService UploadBasket
» MartLink ViewBasket

WebService UploadBasket

See chapter “WebService UploadBasket“

MartLink ViewBasket

The following parameters in the table apply for the MartLink ViewBasket, as well as all general parameters from chapter 4.

URL

The MartLink-ID for ViewBasket is “viewbasket”.

https://‹Portal-URL›/martlink/ viewbasket

Name	Type	Required/ Optional	Description
BasketID	String [128] case sensitive	Required	24 digit UUID of the shopping basket, e.g. ATDBHGBd9iwAAAD2c8Nv.vhn In the GET method escaped e.g. ATDBHGBd9iwAAAD2c8Nv**%2e**vhn
ClientTargetURL	String [1..512] case insensitive	Required	URL of the client to whom the Request is sent.
ClientRequestMethod	String [1..32] [“localget”, ”localpost”]	Required	Method by which the Response to the calling Client is passed.
MartLinkId	String [1..128] case sensitive	Required	Identifier of the calling MartLinks. e.g. ViewBasket

Table 9 *Request Parameter for the MartLink ViewBasket*

Link of Modules

The two modules have to be linked together so that the user can call prices and availability with one click.

The order request is automatically triggered once the shopping basket was sent on the portal by UploadBasket. This function is controlled by the parameter “command”.

The order request is sent to the ERP system. Then the request is edited and returns. After that, the prices and availability are ready for retrieval. The request to collect the shopping basket has to be sent automatically by the computer after a certain time. It is recommended to wait 5 seconds plus 2 seconds for each position before a renewed pick up of the shopping basket is possible.

It is a login of a user of the buyer company needed to display a shopping basket. That means a buyer agent is also able to call a shopping basket which was created by an employee of the same company.

10.4. Online Request of Price and Availability

Another possibility to check price and availability of products is to use the portal interface.

But it is not necessary to change the application to the web browser and to login separately. He can use the direct entry into the portal without logging in by an integrated portal interface into the own computer.

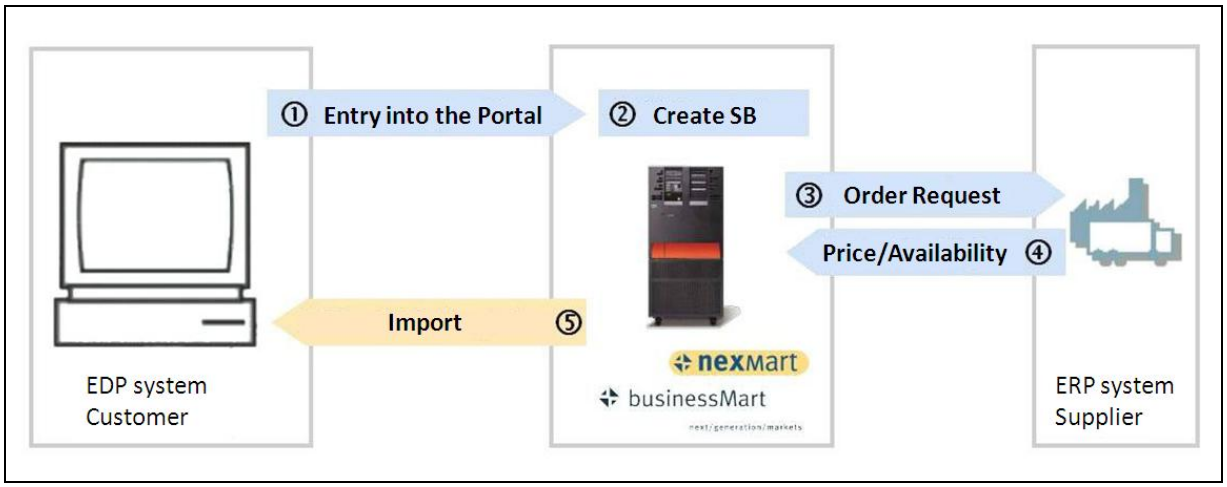

Figure 34 *Schematic representation of the Scenario*

☞ See also “Takeover of Shopping Baskets” and Chapter 6.2 “Entry Point“.

11. Base of all WebServices

11.1.1. Request (for new structure)

Name	Type	Required/ Optional	Description
RequestHeader:ClientId	String [1..128] case sensitive	required	Name of the requested client
buyer:portalId	String [1..128] case insensitive	required	Name of the portal
buyer:organization	String [9] case sensitive	required	ID of the customers company consist 3 letters and 6 digits: e.g. BDE999999
buyer:userName	String [1..255] case sensitive	required	The username of the customer consists of ‹name›.‹surname›. Special characters are not allowed.
buyer:password	String [8..128] case sensitive	required	Password of the user

11.1.2. Response (for new structure)

Name	Type	Guaranteed?	Description
ResponseHeader:ResponseId	String case insensitive	Yes	Unique number of the Response for identification and correlation of internal processes.
results:result	List of result values	Yes	

12. WebService ProcessOrder

Order Process

The WebService ProcessOrder represents the ordering process. The data of a sent order are transformed by the Business Integration Server (BIS). Then the order gets booked in the particular portal. The portal accepts or rejects the order. The acceptance of the order means that the order is accepted by the port and the ordering process was triggered.

Error Handling

Errors of an order can occur in the format or even in the order data itself. Depending on the type of error, tasks are generated either automatically or manually by an administrator.

All processes and tasks are checked for format and order data and displayed by the SCCC (Silent Commerce Control Center).

Format Verification

The data information takes place by means of the Business Integration Sever (BIS). If this fails, the task “format correction” is generated. That happens automatically or manually by an administrator.

Order Data Verification

The order correction is initiated and the task generated if the booking fails. This means that errors were made in the order data, e.g. unknown article number or a delivery date in the past.

Task Pool/SCCC

Every process is controlled by the SCCC. Every task for correction is displayed in the SCCC organizer of a portal. The tasks for corrections may be delegated to the supplier.

The WebService ProcessOrder is used for asynchronous handling of orders. An order is sent in the bM-Order-XML with the necessary authentication and authorization data to the appropriate port by the request of the WebService. The answer includes a ticket for order tracking or an error description. The order confirmation will be sent as an email after the successful processing of the order.

The normal correction process is started and a task generated when an error occurs during the ordering process.

In contrast to the normal order correction is this task delegated to the customer first and only after a specified time is it automatically delegated to the Organizer.

12.1. Service-Routing

The routing of the WebService is performed by the URL and the general WebService request parameter. The following are the relevant parameters with their current values:

» WebService:Id = “ProcessOrder”
» WebService:Version = “1.0”
» WebService:PortalID
» WebService:SupplierId = valid name of the supplier

12.2. URL

URL for the Test-System:
http://soa2.businessmart.de/ProcessOrderWebServiceAdapterRouter/services/ProcessOrder

URL for the Live-System:
http://soa1.businessmart.de/ProcessOrderWebServiceAdapterRouter/services/ProcessOrder

12.3. International Use

The parameters WebService:Locale and WebService:Currency have to be in the Request of the WebService. These parameters control the following:

» All texts of the shopping basket, which are depending on the language, are interpreted as default if no language is specified in the order document. The order confirmation is generated in that language.
» All specifications of the currency in the shopping basket are interpreted as default if nothing else is specified in the Order document. The order confirmation is generated in that currency.

12.4. Security

The transmission path for the request needs to be secured. Also has the authentication and authorization to be ensured for the caller to guarantee the security of the WebService.

https The security of the transmission path is given by the use of the **http(s)-protocol**.

Authentication

The user/password combination is transmitted for authentication. The user is identified by **Buyer:Organization** and **Buyer:UserName** and authorized by **Buyer:UserPassword**. Both have to match an activated and valid ID on the portal.

In addition, the user must have the right to use the **ProcessOrder WebService**.

12.5. Format of the Order

The order returns as bM-OrderXML V1.1 in the answer of theWebService SOAP document.

12.6. Client Information

The name of the client will be transmitted by the parameter Name for statistical purpose. The parameter Client:eMailAdress can transmit information about errors and incidents which cannot be returned in the response of a WebService. A description of the malfunction will be sent at this address.

12.7. Version Management

WebServices will be further developed. The version of the WebService is indicated in the part WebService to run and route one and the same WebService parallel.

12.8. Traceability of the Response

The information of a WebService for tracing is provided in the response. This information could be used for debugging.

The following information is provided:

» **PortalID:**
The portal identification which is generated by the Response.

» **HostName:**
The name of the host which is generated by the response. This information is used only for internal use and tracking errors.

» **TimeStamp:**
A time stamp that holds the date and time of the created Response.

12.9. Exceptions and Error Handling

A result set is transmitted with a summary of the set in the response of the WebServices. The Summary Results:Summary can contain the values "success", "info", "warning", "error" or "fatal".

» **„success“**
Successful execution of the WebService
» **“info“**
Successful execution of a WebService with additional information.
» **“warning“**
Partly successful execution of a WebService with possible different standard conduct. Additional information is provided.
» **“error“**
The execution of the WebService without additional negative impact is not possible. An error description is provided.
» **“fatal“**
Cancellation of a WebService with a negative impact or indeterminate state. An error description is provided.

The Result:Summary presents the highest level of the Result:Level. This shall apply:

“success“ ‹ “info“ ‹ “warning“ ‹ “error“ ‹ “fatal“

The following parameters are transmitted for each entry in the Result-Set:

» **result:code**
The symbolic code of Results is constructed in the following format: “NNNN L CCCC“ [NNNN is the number of the WebService (0000 correlate to general Results), L is the Level (0=success, 1=info, 2=warning, 3= error und 4=fatal) and CCCC is the error code].

» **result:text**
A plaintext message can be sent to the error code in this field. The language is based on the WebService:Locale, the fallback is English.

ProcessOrder response with an error message for *format errors*.

12.10. Technical Implementation

12.10.1. Request Parameter

The following parameters in the table apply for the WebService ProcessOrder, as well as all general parameters from chapter 4.8.1.

Name	Type	Required/ Optional	Description
client:name	String [1..128] case sensitive	required	The name of the called client.
client:emailAddress	String [0..256] case insensitive	optional	The email address of the client for feedback of technical problems.
webService:id	String [1..128] case sensitive	required	The identifier of the calling WebService.
webService:version	String [1..32] case insensitive	required	The Version of the WebService named in the WebServiceID.
webService: portalId	String [1..128] case insensitive	required	Name of the portal.
webService: supplierId	String [1..128] case insensitive	optional	A valid identifier of a supplier.
webService:locale	String [4] case sensitive	required	A valid locale-ID consists three parts: - ISO 639 Language Code 2 letter lower case - separator „_“ - ISO 3166 Country Code 2 letter upper case e.g. de_DE, de_AT, en_GB, ...
webService: currency	String [3] case insensitive	required	A valid currency-ID according to ISO 4217. e.g. EUR, CHF, USD, etc.
webService:tan	Integer	not yet implemented	A valid TransActionNumber

Name	Type	Required/ Optional	Description
buyer:organization	String [9] case sensitive	required	ID of the customers company consist 3 letters and 6 digits: e.g. BDE999999.
buyer:userName	String [1..255] case sensitive	required	The username of the customer consists of ‹name›.‹surname›. Special characters are not allowed.
buyer:password	String [8..128] case sensitive	required	Password of the user.
security:signature	String [1..1024] Case sensitive	not yet implemented	A valid signature on all parts, except the signature itself.
order	bmXML-1.11.xsd	required	The processed order as Order/Basket XML document.

12.10.2. Response

The following parameters in the table apply for the WebService ProcessOrder, as well as all general parameters from chapter 4.8.2.

Name	Type	Required/ Optional	Description
webService:portalId	String [1..128] case insensitive	Yes	Name of the portal
webService:hostName	String [1..256] case sensitive	No	The name of the host which is generated by the response. This information is used only for internal use and tracking errors.
webService: timestamp	DateTime	No	A time stamp that holds the date and time of the created Response.

Name	Type	Required/ Optional	Description
security:signature	String [1..1024] case sensitive	not yet implemented	A valid signature on all parts, except the signature itself.
results:summary attributes of results	String [success, info, warning, error , fatal] case insensitive	Yes	Conclusion of the WebService call, for detailed description see chapter 4.7
result:code attributes of results	String [11]	Yes	The symbolic code of Results is constructed in the following format: “NNNN L CCCC“ [NNNN is the number of the WebService (0000 correlate to general Results), L is the Level (0=success, 1=info, 2=warning, 3= error und 4=fatal) and CCCC is the error code].
result:text	String [0..4096]	Optional	A plaintext message can be sent to the error code in this field. The language is based on the WebService:Locale, the fallback is English.

12.10.3. Results

Result:Level	Result:Code	Result:Text
success	000200000	Order is accepted and will be processed.
error	000230010	Format error: Order document could not be parsed.
info	000011100	The WebService is deprecated and is put out of service in the future. Please contact the provider.
info	000011110	The Version of the WebServices is deprecated. The continued use of the version is not recommended! Please contact the provider.
error	000031100	Client:Name is missing.
error	000032100	WebService is unknown (wrong WebServiceID)
error	000032200	WebService Version unknown.
error	000032300	WebService temporarily out of action.
error	000032400	Unknown portal
error	000032500	Supplier invalid
error	000032600	Locale invalid
error	000032700	Currency invalid
error	000032800	TAN invalid (not yet implemented)
error	000033100	Authentication or authorization of the the specified user has failed (unknown Organization / User combination or wrong password).
error	000033200	Authorization for the called WebService is not available (no rights).

13. FAQ

Error	Possible Cause Proposed Solution
No Basket-ID returned	» BDE/user/password is invalid » supplier (TechnicalSellerName) is invalid (does not exist; capitalization) » customer ID does not exist for this supplier » customer ID is disabled by the supplier » invalid destination country in alternate delivery address » invalid delivery address (e.g. postcode incomplete) » invalid shopping basket-head-fields (commission)
‹result code="000130020"/›	» shopping basket is already picked up » BDE/user/password contain special characters (e.g. äöüß)
Article is not identified	» catalogueArticleCode and ordererArticleCode got mixed up
‹faultcode›Server.generalException‹/faultcode› ‹faultstring›WSWS3277E: Error: No such operation 'ExportBasketRequest'‹/faultstring›	» The request was send to the wrong endpoint
The shopping basket can be picked up after “UploadBasket” and “simulate” but does not change the result code.	» The requested unit is not valid in the portal or ERP system

Table 10 *FAQ’s*

14. Appendix

14.1. Error Codes

The current error codes are available at http://api-test.businessmart.de/services/emartservices/doc/eMartServices_ErrorCodes.html.

MessageCode	Level	MessageText
EMBADR201050	WARNING	Object warning report
EMBADR301010	ERROR	Supplier does not allow any different delivery address
EMBADR301020	ERROR	No further specification of an address error
EMBADR301030	ERROR	The delivery to the specified country is not possible
EMBADR301040	ERROR	Zip code error
EMBADR301050	ERROR	Object error report
EMBADR303340	ERROR	A field is missing in the delivery address
EMBBUY201080	WARNING	Customer ID is disabled in the suppliers system
EMBBUY201090	WARNING	ERP: customer has no read permission
EMBBUY201100	WARNING	ERP: customer has no write permission
EMBBUY201110	WARNING	ERP: customer cannot be executed. Wrong customer ID
EMBBUY201120	WARNING	ERP: customer financially
EMBBUY201140	WARNING	ERP: customer has exceeded credit limit
EMBBUY201150	WARNING	The currency is changed by the ERP of the supplier through the requested customer number
EMBBUY201160	WARNING	Customer has delivery stop in the supplier’s ERP
EMBBUY201170	WARNING	Customer warning with report
EMBBUY301010	ERROR	Customer-Login is unknown in the portal
EMBBUY301020	ERROR	Activation is equivocal or not active at the supplier
EMBBUY301030	ERROR	User has no right to order at the supplier
EMBBUY301040	ERROR	Discount list could not be identified
EMBBUY301050	ERROR	ERP: format error in the customer format. Please correct in the portal
EMBBUY301060	ERROR	ERP: customer ID invalid
EMBBUY301070	ERROR	ERP: unknown error in the customer ID
EMBBUY301080	ERROR	Customer ID is disabled in the suppliers system

MessageCode	Level	MessageText
EMBBUY301090	ERROR	ERP: customer has no read permission
EMBBUY301100	ERROR	ERP: customer has no write permission
EMBBUY301110	ERROR	ERP: customer cannot be executed. Wrong customer ID
EMBBUY301120	ERROR	ERP: customer financially
EMBBUY301130	ERROR	ERP: unknown error in the customer ID
EMBBUY301140	ERROR	ERP: customer has exceeded credit limit
EMBBUY301150	ERROR	The currency is changed by the ERP of the supplier through the requested customer number
EMBBUY301160	ERROR	Customer has delivery stop in the supplier's ERP
EMBBUY301170	ERROR	Customer error with report
EMBBUY301180	ERROR	Field Service: error in registration of the customer
EMBBUY301190	ERROR	Field Service: customer not found
EMBBUY301200	ERROR	Field Service: customer has no right to order at the supplier
EMBBUY301210	ERROR	Field staff has no right to order for the customer
EMBBUY301220	ERROR	Field Service: error in registration of the customer
EMBBUY301230	ERROR	ERP: error in the delivery address
EMBLOC201020	ERROR	Specified language is not supported
EMBOHC201010	WARNING	The supplier has not specified a configurable field
EMBOHC201020	WARNING	No configurable field with this name defined
EMBOHC201030	WARNING	Value of the configurable field incorrectly
EMBOHC201040	WARNING	Multiple warnings for configurable fields
EMBOHC301050	ERROR	Required values for configurable fields do not exist
EMBOHC301060	ERROR	Required value for configurable field does not exist
EMBOHC301070	ERROR	Incorrect value for configurable field
EMBOHC301080	ERROR	Multiple error for configurable fields
EMBOTH303330	ERROR	Error in the order number
EMBPLC201010	WARNING	Product configuration warning with report
EMBPLC301010	ERROR	Product configuration error with report
EMBPLC301020	ERROR	Error in product configuration
EMBPLC301030	ERROR	No product configuration possible
EMBPLP301010	ERROR	Requested currency is not supported

MessageCode	Level	MessageText
EMBPLQ201040	WARNING	Article is currently not available
EMBPLQ201060	WARNING	Article is partly available
EMBPLQ201070	WARNING	Quantity and unit changed by the suppliers system
EMBPLQ201080	INFO	Amount was increased to a minimum order quantity
EMBPLQ201090	INFO	Amount was increased to the next packaging unit
EMBPLQ201110	INFO	Amount was reduced to maximum order quantity
EMBPLQ201130	WARNING	Quantities alert with report
EMBPLQ201140	WARNING	Begin of a set: Next set without a beginning
EMBPLQ201150	WARNING	Quantity and unit changed and rounded by the system of the supplier
EMBPLQ201160	INFO	Availability check for this article is currently not possible
EMBPLQ201170	WARNING	Text fields are truncated when booking in the ERP
EMBPLQ301030	ERROR	The maximum order quantity is exceeded
EMBPLQ301040	ERROR	Article is currently not available and therefore not available for order
EMBPLQ301050	ERROR	Availability check for this article is currently not possible
EMBPLQ301080	ERROR	Error in quantity unit
EMBPLQ301090	ERROR	Error in quantity unit
EMBPLQ301120	ERROR	There are positions with incorrect quantity units
EMBPLQ301130	ERROR	Quantities alert with report
EMBPLQ301170	ERROR	Text fields are truncated when booking in the ERP
EMBPLQ302010	ERROR	Quantity unit or packaging do not match the data of the supplier system
EMBPLQ302020	ERROR	The minimum order quantity was not reached
EMBPLS201020	WARNING	Article in unknown in the ERP
EMBPLS201040	WARNING	Article is currently not available
EMBPLS201050	WARNING	Article is no longer available (discontinued item)
EMBPLS201060	WARNING	Article error ARTICLE_ID_NO_PRICE
EMBPLS201070	WARNING	Article warning with report
EMBPLS201190	WARNING	Text fields are truncated when booking in the ERP
EMBPLS301010	ERROR	ERP: format error in the article number
EMBPLS301020	ERROR	Unknown article number in the system of the supplier
EMBPLS301030	ERROR	ERP: article number error. The reason is not precisely defined

MessageCode	Level	MessageText
EMBPLS301040	ERROR	Article is currently not available
EMBPLS301050	ERROR	Article is no longer available (discontinued item)
EMBPLS301060	ERROR	ERP: Price on request. Please contact your supplier
EMBPLS301070	ERROR	Article error with report
EMBPLS301080	ERROR	Error in specifying a fixed price
EMBPLS301090	ERROR	Error in specifying a fixed price
EMBPLS301100	ERROR	Error in specifying a discount (unit: percent)
EMBPLS301110	ERROR	ERP error in specifying a discount (unit: percent)
EMBPLS301120	ERROR	Error in specifying a discount (unit: percent)
EMBPLS301130	ERROR	Article/currency error
EMBPLS301140	ERROR	Error in article price of the ERP
EMBPLS301150	ERROR	Price error in discount list
EMBPLS301160	ERROR	Currency error
EMBPLS301170	ERROR	General discount list error
EMBPLS301180	ERROR	Discontinued article
EMBPLS301190	ERROR	Text fields are truncated when booking in the ERP
EMBPOS201010	WARNING	Free position, no catalogue article
EMBPOS202030	WARNING	Free position, no catalogue article
EMBPOS301010	ERROR	No catalogue article
EMBPOS301020	ERROR	Position was deleted automatically
EMBPOS301030	ERROR	Error for reorder in order number
EMBPOS301040	ERROR	Error for reorder, position number does not exist
EMBPOS301050	ERROR	Error for reorder, position number is not a number
EMBPOS301060	ERROR	Error for reorder, order number is not a number
EMBPOS301070	ERROR	Error for reorder, position number too long
EMBPOS301080	ERROR	Error for reorder, order number too long
EMBPOS302030	WARNING	Free positions could not be booked because the direct connection to the suppliers system was not available (eMartCode: 32143220)
EMBSEL201060	WARNING	Sales Centre warning with report
EMBSEL201080	WARNING	Supplier does not support the currency in eMart
EMBSEL301010	ERROR	Unknown supplier

MessageCode	Level	MessageText
EMBSEL301020	ERROR	No supplier was given in the order
EMBSEL301030	ERROR	Supplier is offline
EMBSEL301040	ERROR	Supplier does not support the currency in eMart
EMBSEL301050	ERROR	The ERP of the supplier does not support the requested currency
EMBSEL301060	ERROR	Sales Centre error with report
EMBSEL301070	ERROR	The currency of the suppliers ERP is not supported by eMart
EMBSHP201110	WARNING	Format error in the delivery date
EMBSHP201120	WARNING	Invalid delivery date
EMBSHP201130	WARNING	Supplier does not accept the requested delivery date: too early
EMBSHP201140	WARNING	Supplier does not accept the requested delivery date: too late
EMBSHP201150	WARNING	ERP delivery warning with report
EMBSHP301010	ERROR	Error in the shipping method
EMBSHP301020	ERROR	Error in the requested delivery date: format error, date is in the past or too far in the future
EMBSHP301030	ERROR	Requested delivery date is too far in the future
EMBSHP301040	ERROR	ERP: format error in the delivery date
EMBSHP301050	ERROR	ERP: invalid delivery date
EMBSHP301060	ERROR	Supplier does not accept the requested delivery date: too early
EMBSHP301070	ERROR	Supplier does not accept the requested delivery date: too late
EMBSHP301080	ERROR	Date is mandatory but was not specified
EMBSHP301090	ERROR	Location is mandatory but was not specified
EMBSHP301100	ERROR	Shipping method is not supported by the supplier
EMBSHP301110	ERROR	Format error in the delivery date
EMBSHP301120	ERROR	Invalid delivery date
EMBSHP301130	ERROR	Supplier does not accept the requested delivery date: too early
EMBSHP301140	ERROR	Supplier does not accept the requested delivery date: too late
EMBSHP301150	ERROR	ERP delivery error with report
EMBSHP301160	ERROR	You must specify a date in this shipping method!
EMBSHP301170	ERROR	The supplier requires a date that is at least X days in the future!
EMBSHP301180	ERROR	The supplier requires a date that is more than X days in the future!
EMBSHP301190	ERROR	Please enter a date with a valid format!

MessageCode	Level	MessageText
EMBSHP301200	ERROR	You must specify a date and a place for this shipping method
EMBSHP303320	ERROR	Error in the validation of the delivery date
EMBWKF101010	SUCCESS	CreateResult successful
EMBWKF101050	SUCCESS	SimulateResult successful
EMBWKF101090	SUCCESS	SubmitResult successful
EMBWKF101130	SUCCESS	ConfirmResult successful
EMBWKF101170	INFO	Final response on the successful completion of the process. Business objects have to be appended to the response
EMBWKF201020	WARNING	CreateResult warning
EMBWKF201060	WARNING	SimulateResult warning
EMBWKF201100	WARNING	SubmitResult warning
EMBWKF201140	WARNING	ConfirmResult warning
EMBWKF201190	WARNING	Order could not be found (for the interface RX_GetOrderController)
EMBWKF201210	INFO	Error in the order positions found
EMBWKF201240	WARNING	Warning: order request in the ERP contains errors
EMBWKF301010	ERROR	Free positions are not allowed
EMBWKF301030	ERROR	CreateResult error
EMBWKF301070	ERROR	SimulateResult error
EMBWKF301110	ERROR	SubmitResult error
EMBWKF301150	ERROR	Error on sending order confirmation. Please check e-mail address
EMBWKF301200	ERROR	A duplicate has been sent from your ERP system. An order was already processed at this number.
EMBWKF301210	ERROR	Shopping basket for the BasketID not found
EMBWKF301220	ERROR	The customer is not the owner of the shopping basket for the BasketID
EMBWKF301230	ERROR	Shopping basket is ready for export
EMBWKF301240	ERROR	Order request in the ERP contains errors
EMBWKF301250	ERROR	Data error, necessary data are missing for the order request at the ERP
EMBWKF301260	ERROR	Data error, necessary data are missing for the order request at the ERP

Table 11 *Error Codes*

bmXML-ErrorCode	eMart-ErrorCode	ErrorLevel	Reference	Description
EMBWKF101170	11011020	info		Final answer about success of processing a request. The Business objects must be attached to the response.
EMBWKF201240	22500040	Warning	//basketId	Basket not synchronized successfully with ERP for export
EMTFMT302030	31901130	Error		Error transforming data into eXML
EMTSEC301010	31901210	Error		Authorization error. Happens when external component can provides incorrect login and password during POST request.
EMBWKF301210	32500010	Error	//basketId	Basket not found for BasketID.
EMBWKF301220	32500020	Error	//buyerName	Buyer is not owner of Basket for BasketID.
EMBWKF301230	32500030	Error	//basketId	BasketStatus wrong for export
EMBWKF301240	32500040	Error	//basketId	Basket not synchronized successfully with ERP for export
EMTSYS403050	41901900	Fatal		General error. Uncontrollable exception.

Table 12 *Error Codes*

LoginError	Password is incorrect
LoginInvalid	User is incorrect or does not exist
BuyerCompanyNotAuthorized	Company is not authorized for this MartService
BuyerNotAllowed	User is not activated for this MartService
UknownBuyerCompany	Name of company is incorrect or unknown
Supplier	Supplier is incorrect
SellerCompanyNotFound	Name of the supplier is incorrect
SellerNotAllowed	Supplier is not activated for this MartLink
InvalidForSellerCompany	
LocaleError	Country name is not correct, for example de_DE
StoreFrontModeError	
StoreClientTargetURLError	
GeneralError	General/Unknown error. Please contact your Support-Team.
DetermineProductFailed	Unknown article number (only for MartLink ProductDetail)

Table 13 *Error Codes*

14.2. VBA Example Shopping Basket Takeover

VBA Example Shopping Basket Takeover

```
Option Compare Database
Option Explicit



Private Sub cmdBack_Click()

' Navigate back
On Error Resume Next
ocxWeb.GoBack

End Sub



Private Sub cmdForward_Click()

' Navigate forward
On Error Resume Next
ocxWeb.GoForward

End Sub



Private Sub cmdLaunch_Click()

' Navigate to Mart links page
ocxWeb.Navigate txtLinkPage

End Sub
```

VBA Example Shopping Basket Takeover

```
Private Sub cmdLaunchService_Click()


' Generate SOAP Envelope
Dim stXML As String
stXML = funcFormatXML("<?xml version=""1.0"" encoding=""UTF-8""?>", 0)
stXML = stXML & funcFormatXML("<soapenv:Envelope
xmlns:soapenv=""http://schemas.xmlsoap.org/soap/envelope/"" xmlns:xsd =
""http://www.w3.org/2001/XMLSchema"" xmlns:xsi=""http://www.w3.org/2001/XMLSchema-instance"">", 0)
stXML = stXML & funcFormatXML("<soapenv:Body>", 1)
stXML = stXML & funcFormatXML("<ExportBasketRequest xmlns="""">", 2)
stXML = stXML & funcFormatXML("<client>", 3)
stXML = stXML & funcFormatXML("<name>Test-Client</name>", 4)
stXML = stXML & funcFormatXML("<emailAddress>michael.mezger@businessmart.de</emailAddress>", 4)
stXML = stXML & funcFormatXML("</client>", 3)
stXML = stXML & funcFormatXML("<webService>", 3)
stXML = stXML & funcFormatXML("<id>GetBasket</id>", 4)
stXML = stXML & funcFormatXML("<version>1.0</version>", 4)
stXML = stXML & funcFormatXML("<portalId>" & txtPortalID & "</portalId>", 4)
stXML = stXML & funcFormatXML("<supplierId>" & txtSupplierID & "</supplierId>", 4)
stXML = stXML & funcFormatXML("<locale>de_DE</locale>", 4)
stXML = stXML & funcFormatXML("<currency>EUR</currency>", 4)
stXML = stXML & funcFormatXML("<tan>" & txtTAN & "</tan>", 4)
stXML = stXML & funcFormatXML("</webService>", 3)
stXML = stXML & funcFormatXML("<buyer>", 3)
stXML = stXML & funcFormatXML("<organization>" & txtBDE & "</organization>", 4)
stXML = stXML & funcFormatXML("<userName>" & txtLogin & "</userName>", 4)
stXML = stXML & funcFormatXML("<password>" & txtPwd & "</password>", 4)
stXML = stXML & funcFormatXML("</buyer>", 3)
stXML = stXML & funcFormatXML("<security/>", 3)
stXML = stXML & funcFormatXML("<basketId>" & txtBasketId & "</basketId>", 3)
stXML = stXML & funcFormatXML("<check>", 3)
stXML = stXML & funcFormatXML("<checkSum>" & txtCheckSum & "</checkSum>", 4)
stXML = stXML & funcFormatXML("</check>", 3)
```

VBA Example Shopping Basket Takeover

```
stXML = stXML & funcFormatXML("‹/ExportBasketRequest›", 2)
stXML = stXML & funcFormatXML("‹/soapenv:Body›", 1)
stXML = stXML & funcFormatXML("‹/soapenv:Envelope›", 0)

' Display SOAP Message
txtRequest = Replace(stXML, vbTab, "    ")

' Transfer SOAP Message to SOA
Dim xmlRequest As New XMLHTTP
With xmlRequest
   .Open "POST", txtWebServiceURL, False, txtLogin, txtPwd
   .setRequestHeader "Content-Type", "text/xml"  'Bei Fehlern den Wert "application/x-www-form-urlencoded" für Content-Type verwenden
   .setRequestHeader "SOAPAction", "getBasket"
   .send stXML
End With

' Display response
txtResult = Replace(xmlRequest.responseText, "›‹", "›" & vbCrLf & "‹")

End Sub

Private Sub cmdRestore_Click()

' Call form load event
Call Form_Load

' Empty text boxes
txtSupplierID = ""
txtCheckSum = ""
txtBasketId = ""
txtTAN = ""
txtBDE = "BDE999999"
```

VBA Example Shopping Basket Takeover

```
txtLogin = "Admin.Admin"
txtPwd = "passwort"
txtPortalID = "demomart.de"

' Clear web component
'ocxWeb.Navigate ""
'ocxWeb.Navigate txtLinkPage
ocxWeb.Navigate txtLocalHTML

End Sub

Private Sub Form_Load()

' Fill text boxes with default settings
txtLocalHTML = "file:///c:/Temp/test.html"
txtLinkPage = "http://demomart.businessmart.de/links.html#MartLinkPickBasket"
txtWebServiceURL = "http://swa2.businessmart.de/GetBasketAdapterRouter/services/ExportBasket"
txtBDE = "BDE999999"
txtLogin = "Admin.Admin"
txtPwd = "passwort"
txtPortalID = "demomart.de"

End Sub

Private Sub ocxWeb_NavigateComplete2(ByVal pDisp As Object, URL As Variant)

' Variable declaration
Dim intCounter As Integer

' Check if test page has been loaded
If InStr(1, URL, Mid(txtLocalHTML, InStrRev(txtLocalHTML, "//") + 2)) › 0 Then
   ' Get parameter string
   Dim stParameters As String
```

VBA Example Shopping Basket Takeover

```
    stParameters = Mid(URL, InStr(1, URL, "?") + 1)

    ' Split parameters into array
    Dim arrParameters() As String
    arrParameters = Split(stParameters, "&")

    ' Get parameters and their corresponding values
    Dim arrParamValues() As String
    Dim stResult As String
    For intCounter = 0 To UBound(arrParameters) - 1
      arrParamValues() = Split(arrParameters(intCounter), "=")
      Select Case arrParamValues(0)
        Case "SupplierId":
          txtSupplierID = arrParamValues(1)
        Case "BasketId":
          txtBasketId = arrParamValues(1)
        Case "CheckSum":
          txtCheckSum = arrParamValues(1)
        Case "TAN":
          txtTAN = arrParamValues(1)
      End Select
      stResult = stResult & arrParamValues(0) & ": " & arrParamValues(1) & vbCrLf
    Next

    ' Display result
    MsgBox stResult, vbInformation, "Übermittelte Parameterwerte"
End If

End Sub
```

14.3. Visual Basic .NET Example ProcessOrder

Visual Basic .NET Example ProcessOrder

```
Public Class Form1
    'URL für den Web Service Aufruf festlegen
    Dim WSTargetUploadBasketURL As String =
"http://soa2.businessmart.de/ProcessOrderWebServiceAdapterRouter/services/ProcessOrder"
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button1.Click

        Dim requestString As String = My.Computer.FileSystem.ReadAllText("bsp_order.xml")

        'MsgBox(requestString)

        Dim responseString As String = ""

        responseString = funcSendSoapRequest(WSTargetUploadBasketURL, requestString, "processOrder")

        responseString = responseString.Replace("><", ">" & Chr(13) & Chr(10) & "<")

        My.Computer.FileSystem.WriteAllText("uploadbasket_response.xml", responseString, False)

        MsgBox("hallo ich bin fertig")

    End Sub

    Private Function funcSendSoapRequest(ByVal url As String, ByVal postContent As String, ByVal soapActions
As String) As String

        Dim httpRequest As Net.HttpWebRequest = Net.HttpWebRequest.Create(url)

        'Proxy settings
        'Dim proxy As Net.WebProxy
        'Dim cred As Net.NetworkCredential
        'proxy = New Net.WebProxy("http://localhost:8888", False)
```

Visual Basic .NET Example ProcessOrder

```
        'cred = New NetworkCredential("proxyUser", "proxyPWD", "proxyDomain")
        'proxy.Credentials = cred
        'httpRequest.Credentials = cred
        'httpRequest.Proxy = proxy

        'set HEADERs
        httpRequest.Method = "POST"
        httpRequest.ContentType = "text/xml"
        httpRequest.Headers.Add("SOAPAction", soapActions)

        'set timout
        httpRequest.Timeout = 120000

        ' Write to request
        Dim requestStream As IO.Stream = httpRequest.GetRequestStream()
        Dim requestStreamWriter As New IO.StreamWriter(requestStream, New System.Text.UTF8Encoding(False))
        requestStreamWriter.Write(postContent)
        requestStreamWriter.Close()
        requestStream.Close()

        ' Open response
        Dim httpResponse As Net.HttpWebResponse
        'httpResponse = CType(httpRequest.GetResponse, Net.HttpWebResponse)
        httpResponse = httpRequest.GetResponse
        Dim responseStream As IO.Stream = httpResponse.GetResponseStream()

        ' Read the response stream.
        Dim streamReader As IO.StreamReader = New IO.StreamReader(responseStream)
        Dim sText As String = streamReader.ReadToEnd()
        Return sText
    End Function
End Class
```